DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93943

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

THE EFFECTIVENESS OF HEAT EXCHANGERS WITH ONE SHELL PASS AND FIVE TUBE PASSES

by

Lawrence E. Hess

September 1985

Thesis Advisor: Allan D. Kraus

Approved for public release; distribution is unlimited

T222823

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) The Effectiveness of Heat Exchangers with One Shell Pass and Five Tube Passes		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; September 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Lawrence E. Hess		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943-5100		12. REPORT DATE September 1985
		13. NUMBER OF PAGES 121
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Heat Exchanger
Effectiveness N_{tu} Method
Five Tube Pass

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Heat exchangers with one shell pass and n tube passes are often referred to as 1-n exchangers. The heat transfer literature contains many references to studies of 1-n exchangers when n is even but apparently, other than a single study and some work pertaining to the 1-3 exchanger at the Naval Postgraduate School, little work has been done with respect to the 1-n exchanger when n is odd. This thesis expands upon the study of 1-n exchangers with n being odd.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S N 0102- LF- 014- 6601

20. (Continued)

While a completely closed form solution was found to be unfeasible, a polynomial approximation has been developed that yields the effectiveness (ε) of the two possible arrangements of the 1-5 exchanger as a function of the capacity rate ratio (R) and the number of transfer units (N_{tu}). This will enable the analyst to consider exchangers where the inlet to and outlet from the tubes are at opposite ends of the exchanger.

Approved for public release; distribution is unlimited

The Effectiveness of Heat Exchangers
With One Shell Pass and
Five Tube Passes

by

Lawrence E. Hess
Lieutenant Commander, United States Navy
B.S., The Citadel, 1973

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1985

Thesis
H5232
c.1

ABSTRACT

Heat exchangers with one shell pass and n tube passes are often referred to as 1-n exchangers. The heat transfer literature contains many references to studies of 1-n exchangers when n is even but apparently, other than a single study and some work pertaining to the 1-3 exchanger at the Naval Postgraduate School, little work has been done with respect to the 1-n exchanger when n is odd. This thesis expands upon the study of 1-n exchangers with n being odd.

While a completely closed form solution was found to be unfeasible, a polynomial approximation has been developed that yields the effectiveness (ε) of the two possible arrangements of the 1-5 exchanger as a function of the capacity rate ratio (R) and the number of transfer units (N_{tu}). This will enable the analyst to consider exchangers where the inlet to and outlet from the tubes are at opposite ends of the exchanger.

TABLE OF CONTENTS

I. INTRODUCTION 13

II. THE DEVELOPMENT OF THE EFFECTIVENESS METHOD 16

A. HISTORICAL DEVELOPMENT 16

III. AN ATTEMPT AT AN ANALYTIC SOLUTION 22

A. EFFECTIVENESS AS A FUNCTION OF CAPACITY RATE RATIO AND EXCHANGER SIZE (N_{tu}) 22

B. ANALYTICAL DEVELOPMENT 24

IV. NUMERICAL AND COMPUTER ANALYSIS 40

A. THERMAL ANALYZER TVSSI 40

B. DEVELOPMENT OF THE 1-5:3C AND 1-5:3P HEAT ECHANGER MODELS 41

C. SCOPE OF COMPUTER ANALYSIS 43

D. INTERPRETATION OF GRAPHS 45

V. POLYNOMIAL REGRESSION 47

A. DEVELOPMENT OF POLYNOMIAL EQUATIONS 47

B. NUMERICAL EXAMPLE 53

VI. CONCLUSIONS AND RECOMMENDATIONS 62

A. CONCLUSIONS 62

B. RECOMMENDATIONS 63

APPENDIX A: NTU53C COMPUTER GENERATED INPUT ANALYZER PROGRAM 64

APPENDIX B: NTU53P COMPUTER GENERATED INPUT ANALYZER PROGRAM 69

APPENDIX C: SAMPLE OUTPUT FROM NTU53 COMPUTER INPUT ANALYZER PROGRAM 74

APPENDIX D: MODIFIED NTU53C PROGRAM USED TO RUN ON BATCH SYSTEM 83

APPENDIX E: MASTER EXECUTION PROGRAM FOR NTU53C AND TVCOUNT 87

APPENDIX F: MODIFIED NTU53C PROGRAM USED TO RUN ON BATCH SYSTEM 91

APPENDIX G: MASTER EXECUTION PROGRAM FOR NTU53P AND TVCOUNT 95

APPENDIX H: 1-5:3P EFFECTIVENESS VS. N_{tu} GRAPHS AT VARIOUS R VALUES 99

APPENDIX I: 1-5:3C EFFECTIVENESS VS. N_{tu} GRAPHS AT VARIOUS R VALUES 102

APPENDIX J: EFFECTIVENESS COMPUTER GENERATED ANALYZER PROGRAM 105

LIST OF REFERENCES 119

INITIAL DISTRIBUTION LIST 121

LIST OF TABLES

1. 1-5:3C COEFFICIENTS 57

2. 1-5:3P COEFFICIENTS 59

3. COMPARISON OF EXAMPLE RESULTS 61

LIST OF FIGURES

1.1 A 1-2 Heat Exchanger (One Shell Pass and Two Tube Passes) 14

1.2 1-5 Heat Exchanger (One Shell Pass and Five Tube Passes) 15

2.1 1-5:3C Five Tube Passes - Three in Counterflow . . 21

2.2 1-5:3P Five Tube Passes - Three in Parallel Flow . 21

3.1 Three Counter Two Parallel Pass Configuration for Development of the Sought After Effectiveness Relationship 25

4.1 Comparison of Analytical (1-4) and Computer Results (1-3:2C and 1-3:2P) to Computer Results (1-5:3C and 1-5:3P) at R = 0.5 46

5.1 1-5:3C Data Fit by a 5th Order Polynomial 51

5.2 1-5:3P Data Fit by a 5th Order Polynomial 52

H.1 1-5:3P Effectiveness vs. N_{tu} over Range of R from 0.01 to 1.0 99

H.2 1-5:3P Effectiveness vs. N_{tu} over Range of R from 0.01 to 0.5 100

H.3 1-5:3P Effectiveness vs. N_{tu} over Range of R from 0.5 to 1.00 101

I.1 1-5:3C Effectiveness vs. N_{tu} over Range of R from 0.01 to 1.0 102

I.2 1-5:3C Effectiveness vs. N_{tu} over Range of R from 0.01 to 0.5 103

I.3 1-5:3C Effectiveness vs. N_{tu} over Range of R from 0.5 to 1.00 104

NOMENCLATURE

English Letter Symbols

A = Exchanger heat-transfer surface, sq m

A_m = Coefficient of the mth value dimensionless

A_0 = Coefficient, dimensionless

A_1 = 1st order coefficient to be multiplied by N_{tu}, dimensionless

A_2 = 2nd order coefficient to be multiplied by N_{tu}^2, dimensionless

A_3 = 3rd order coefficient to be multiplied by N_{tu}^3, dimensionless

A_4 = 4th order coefficient to be multiplied by N_{tu}^4, dimensionless

A_5 = 5th order coefficient to be multiplied by N_{tu}^5, dimensionless

a = Exchanger heat-transfer surface, sq m/m

C = Capacity rate, W/°K. Also designates dimensionless arbitrary constant

C_{pc} = Specific heat at constant pressure of cold fluid, J/kg°K

C_{ph} = Specific heat at constant pressure of hot fluid, J/kg°K

D = Empirical value of effectiveness (computer generated), dimensionless

e = Error. Also used as the exponential function

F = Logarithmic mean temperature difference correction factor, dimensionless

L = Exchanger length, m

N = Number of effectiveness empirical data points used to determine a curve for R, dimensionless

n = Number of tube passes, dimensionless. Also number of equations, dimensionless

n_c = A related N_{tu} per unit length cold side, m^{-1}

n_h = A related N_{tu} per unit length hot side, m^{-1}

N_{tu} = Number of transfer units, dimensionless

P = Temperature group, dimensionless

q = Total rate of heat transfer, W

q_{max} = Maximum total rate of heat transfer, W

R = Capacity rate ratio, dimensionless

S = Temperature group, dimensionless

S_r = Sum of the squares of the residuals, dimensionless

T = Hot fluid temperature, °C

T_{pi} = Particular integral, dimensionless

T_1 = Hot fluid temperature in, °C

T_2 = Hot fluid temperature out, °C

t_1 = Cold fluid temperature in, °C

t_2 = Cold fluid temperature out, °C

t_a = Cold fluid temperature 1st pass, °C

t_{ab} = Cold fluid temperature between 1st and 2nd passes

t_b = Cold fluid temperature 2nd pass, °C

t_{bc} = Cold fluid temperature between 2nd and 3rd passes

t_c = Cold fluid temperature 3rd pass, °C

t_{cd} = Cold fluid temperature between 3rd and 4th passes

t_d = Cold fluid temperature 4th pass, °C

t_{de} = Cold fluid temperature between 4th and 5th passes

t_e = Cold fluid temperature 5th pass, °C

U = Overall heat transfer coefficient, W/m^2 - °C

W = Mass flow, kg/sec.

x = length coordinate, m. Also used to represent a constant value in a sequence

y = Sum of mth degree polynomial, defined by eq. (5.1), dimensionless

Greek Letter Symbols

α = Root of auxiliary differential equation, 1/m

ε = Exchanger effectiveness, dimensionless

λ = Combination of variables defined by equation (3.12), dimensionless

Σ = Summation, dimensionless

σ^2 = Variance, dimensionless

θ_m = Mean temperature difference for exchanger, °C

∂ = Indicates partial derivative, dimensionless

β = A combination of terms defined by eq. (3.23)

Subscripts

c = Cold fluid

h = Hot fluid

i,j,k = Values in a sequence

m = Degree or order, an exponent

1 = inlet

2 = outlet

ACKNOWLEDGEMENT

I wish to thank my wife, Cheri, who has been a sustaining influence from the beginning and to my parents who launched my career. Also my sincere gratitude to Dr. Allan D. Kraus for his time, guidance and patience. Without his assistance and encouragement this thesis would not have been possible.

Finally, I wish to thank Uncle Sam who made one of my life-long goals possible.

I. INTRODUCTION

In many engineering applications, it is desirable to transfer thermal energy between fluids at different temperatures. This action may be economically accomplished with the aid of a heat exchanger.

The primary function of a non-direct contact heat exchanger is to provide two paths of flow, one for the hot fluid and one for the cold, by means of which heat can be transferred through walls separating the fluids.

The quantity of heat transferred is governed by three factors: 1) the amount and nature of the heat transfer surface to which the two fluids are exposed, 2) the overall heat transfer coefficient, and 3) the mean temperature difference across the intervening wall from one fluid to the other.

Numerous flow arrangements and geometric designs are possible. One of the most common types of heat exchangers is the shell and tube type. Here one fluid flows through the tubes. The tubes are enclosed in a shell with provisions for the other fluid to flow through the spaces between and exterior to the tubes. Overall flow may be either parallel or counter, but in most cases, the latter is preferred. The reason for this will be seen later. One of the most common flow arrangements used in the shell and tube heat exchangers is a combination of the two, the parallel-counterflow exchanger. Heat exchangers with one shell pass and n tube passes are therefore referred to as 1-n heat exchangers.

One example of this flow arrangement is the one shell-pass two tube pass exchanger as shown in Figure 1.1.

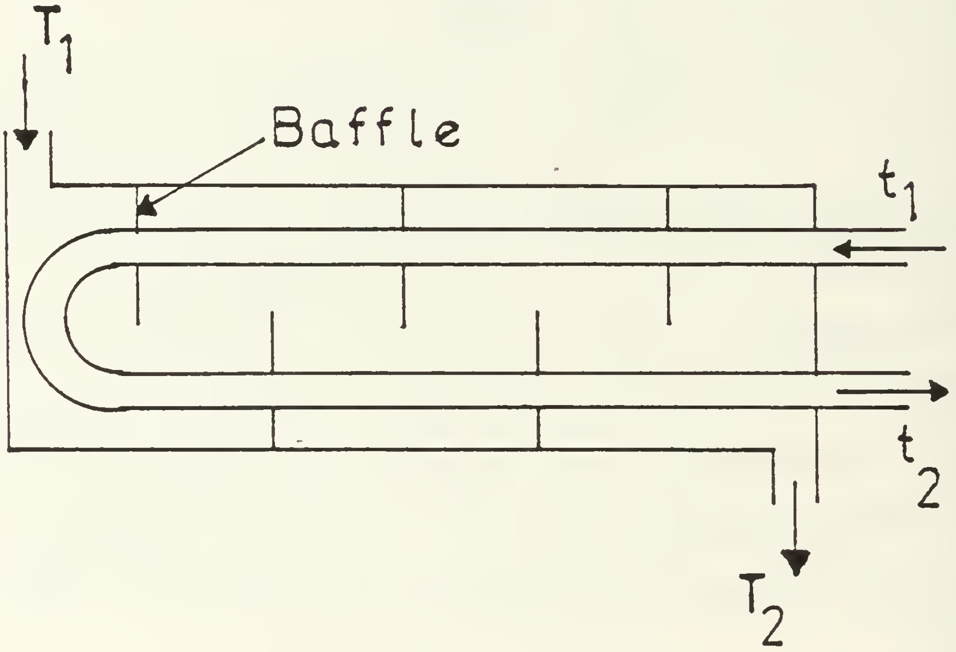

Figure 1.1 A 1-2 Heat Exchanger (One Shell Pass and Two Tube Passes)

This type of exchanger is configured so that all of the tube side fluid flows consecutively through half of the total number of tubes: The baffles in the shell serve to "mix" the shell fluid and to maintain fluid flow at right angles to the tube passes in order to obtain a maximum "shell side" heat transfer coefficient.

O'Hare [Ref. 1] has indicated that, to date, much work has been done on finding the logarithmic mean temperature difference for heat exchangers with an even number of passes but little has been done with exchangers having an odd number of passes. From this basic statement, he proceeded with a theoretical examination of the 1-3 heat exchanger which resulted in some startling conclusions. As a follow up to his work, this thesis will parallel and follow the format of O'Hare [Ref. 1] for the 1-5 parallel counter flow exchanger. One arrangement of the 1-5 exchanger is displayed in Figure 2.

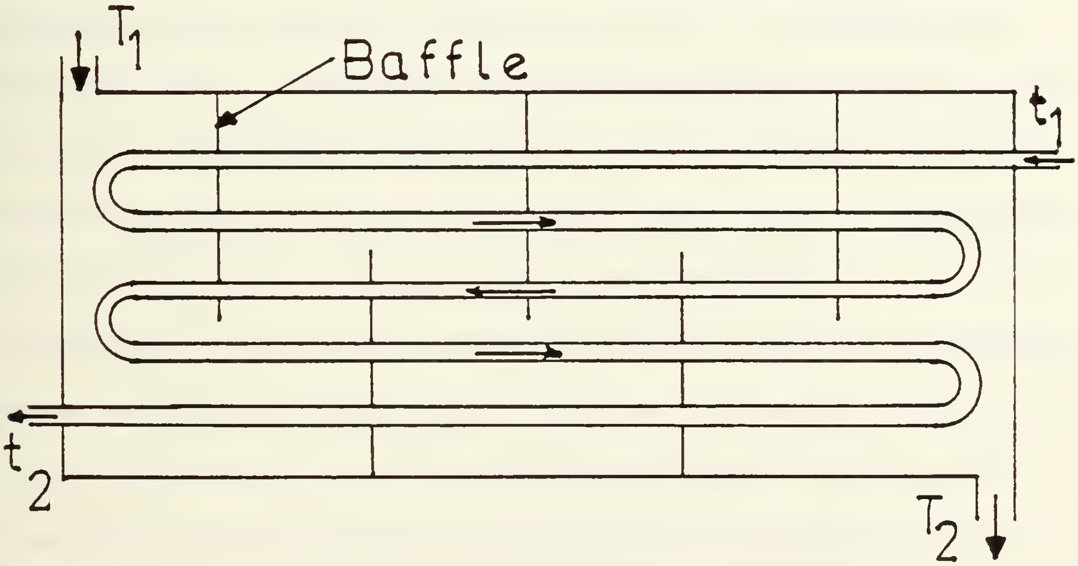

Figure 1.2 A 1-5 Heat Exchanger (One Shell Pass and Five Tube Passes)

II. THE DEVELOPMENT OF THE EFFECTIVENESS METHOD

A. HISTORICAL DEVELOPMENT

Nagle [Ref. 2: pp. 604-609], in 1931, credited Davis [Ref. 3] with a simplified method for computing actual temperature differences between two heat-interchanging streams which depart from true counter or concurrent (parallel) flow. This is now the familiar "F factor" method which expresses the actual mean temperature difference θ_m in $q = UA\theta_m$ as a fraction F of the counterflow logarithmic mean temperature difference, LTMD, θ_{mc} via $\theta_m = F\theta_{mc}$.

The example of initial interest was the 1-2 exchanger with a single shell pass and two continuous tube passes in counter and concurrent flow with it. The method involved derivation of the actual temperature difference for the flow pattern and formed the ratio $F \doteq \theta_m/\theta_{mc}$. This familiar LMTD correction factor was plotted conveniently as functions of the effectiveness, ε, and the capacity rate ratio R with R as a parameter. These mean temperature difference correction charts are available for many flow arrangements [Ref. 4: pp. 829-833 and Ref. 5]. The effectiveness, ε (often called P or S), is always the cold fluid effectiveness and R is always the capacity rate ratio of cold fluid to hot fluid.

Nagle detailed assumptions and derivations for the 1-2, 1-4 and 1-6 exchangers. The F factors were obtained by Nagle through graphical integration and were accompanied by the comment that F factors for the 1-2 exchanger could be applied with negligible error to 1-4 and 1-6 exchangers. Underwood [Ref. 6: pp. 145-148] rederived the equations of Nagle for 1-2 and 1-4 exchangers to eliminate the need for obtaining F factors by graphical integration.

Bowman [Ref. 7: pp. 541-544] pointed out that for a very large or infinite number of tube passes, the F factor approached, as a limit, its value in crossflow with both fluids completely mixed. It was further stated that even at the limit, the F factors were only 1 to 2 percent lower than those for the 1-2 exchanger. A previous paper by Kraus and Kern [Ref. 8] did not confirm the generalization that 1-n exchangers differed only negligibly from the 1-2 exchanger although this lack of confirmation was obtained on an $\varepsilon = f(R, N_{tu})$ basis. Moreover, the Kraus-Kern work does not confirm the generalizations on an $F = f(R, N_{tu}, \varepsilon)$ basis.

From the standpoint of usefulness and good accuracy, it is essential that F factors, if they are to be used in preference to $\varepsilon =$ f(R, N_{tu}, flow arrangement), be obtained with precision. Plots of $F = f(R, \varepsilon, = P \text{ or } S)$ [Ref. 4: pp. 829-833 and Ref. 5] show that the curves for particular values of R approach infinite slope as F decreases. While this can be partially alleviated by restricting $R < 1.0$ (a constraint used in the $\varepsilon = f(R, N_{tu}$, flow arrangement) approach), it is seen that small errors in the interpolation for R or $\varepsilon = P$ or S can result in large fluctuations in the value of F.

In a comprehensive paper, Bowman, Mueller and Nagle [Ref. 9: pp. 283-294] presented graphs of F factors for shells with one through six shell passes and numbers of continuous tube passes respectively double the number of shell passes. In view of the earlier references to Nagle and Bowman, it should be noted that F factors were computed for the 1-2 exchanger in [Ref. 9: pp. 283-294] using the equations of Underwood [Ref. 6: pp. 145-158].

Ten Broeck [Ref. 10: pp. 1041-1042] prepared a graph of the dimensionless groups now known as ε, R and N_{tu} for the 1-2 exchanger. Such a graph had the added versatility of simplifying the calculation of performance in a given exchanger when operating at conditions different for those for which it was designed. Kays and London [Ref. 11: pp. 63-74] prepared similar graphs and tables of $\varepsilon = f(R, N_{tu}$, flow arrangement) for the 1-2 exchanger and for several cases of crossflow and periodic flow.

In the foregoing discussion, it has been observed that there are two methods for the design and analysis of a heat exchanger. These are the so-called "F factor" and "ε-N_{tu}" approaches and it is to be noted that the Heat Transfer Literature contains many references of design equations for either of these methods for exchangers that have one shell pass and even numbers of tube passes.

It was O'Hare [Ref. 1] who pointed out that there may be a virtue in having an odd number of tube passes particularly in a marine application. Such an arrangement allows the tube side fluid to enter one side of the exchanger and leave from the opposite side (see Figure 1.2).

With the possible use for an odd tube pass heat exchanger established, a literature search was conducted to uncover any previously solved cases of 1-n heat exchangers with n, the number of odd tube passes. O'Hare [Ref. 1], Fischer [Ref. 12], and Stevens, et. al. [Ref. 13] have dealt with this type of exchanger. Although Stevens, et. al. deal with multiple crossflow passes, all provide the foundation for the work reported here as well as additional work pertaining to the 1-3 exchanger. O'Hare's investigation centered around the development of a solution for the 1-3 exchanger in the effectiveness-N_{tu} framework while Fischer derived an equation using the mean temperature difference for the 1-3 exchanger. Moreover, Fischer expressed his results in terms of the correction factor F and treats only the case of the 1-3 exchanger. This thesis will expand on the N_{tu} method of thermal heat exchanger design by covering the 1-5 heat exchanger. The work will treat both the three counterflow and two parallel flow and the two counterflow and three parallel flow (1-5:3P) cases (see Figures 2.1 and 2.2).

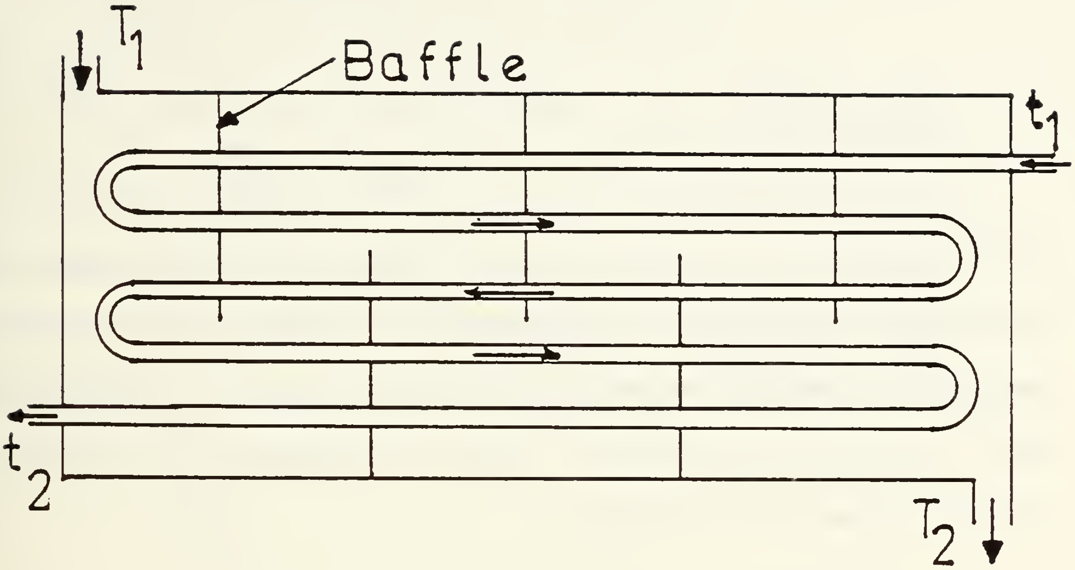

Figure 2.1 1-5:3C Five Tube Passes - Three in Counterflow

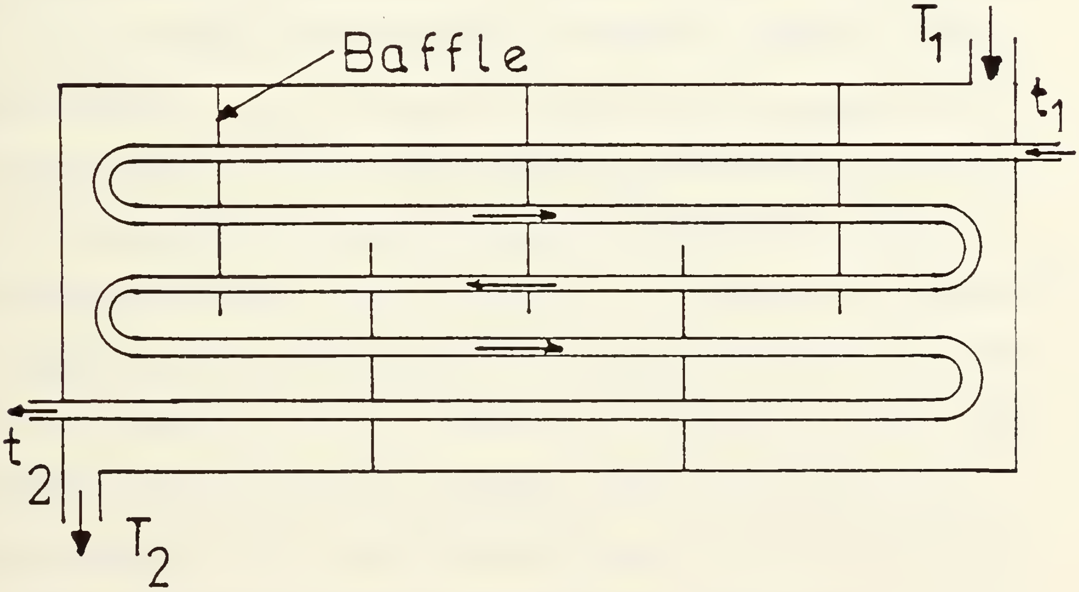

Figure 2.2 1-5:3P Five Tube Passes - Three in Parallel Flow

III. AN ATTEMPT AT AN ANALYTIC SOLUTION

A. EFFECTIVENESS AS A FUNCTION OF CAPACITY RATE RATIO AND EXCHANGER SIZE (N_{tu})

This section deals with an investigation into the effectiveness, ε, of a one shell pass-five tube pass heat exchanger, in which ε compares the actual heat transfer capability to the thermodynamically limited, maximum possible heat transfer capability. This exchanger heat transfer effectiveness is given by

$$\varepsilon = \frac{q}{q_{max}} = \frac{C_h(T_{hot,in} - T_{hot,out})}{C_{min}(T_{hot,in} - t_{cold,in})} = \frac{C_c(t_{cold,out} - t_{cold,in})}{C_{min}(T_{hot,in} - t_{cold,in})}$$

where C_{min} is the smaller of the C_h and C_c magnitudes (the capacity rates). Thus, ε possesses the significance of effectiveness of the heat exchanger from a thermodynamic point of view, with the magnitude of the effectiveness completely defining the heat transfer performance. In general we express $\varepsilon = f(N_{tu}$, R, and flow arrangement) and when the flow arrangement is specified or understood, it is said that $\varepsilon = f(N_{tu}, R)$. [Ref. 11: pp. 14-26].

The number of heat transfer units N_{tu} is a nondimensional expression of the "heat transfer size" of the exchanger. When N_{tu} is small the exchanger effectiveness is low, and when N_{tu} is large, ε asymptotically approaches the limit imposed by the flow arrangement and thermodynamic conditions. From inspection of the definition of N_{tu}

$$N_{tu} = \frac{AU}{C_{min}} = \frac{1}{C_{min}} \int_0^A UdA$$

it is clear that the overall conductance and transfer area affect the costs of attaining a high value for N_{tu}, ergo a high ε. The capacity rate ratio, R, as defined by

$$R = \frac{C_{min}}{C_{max}}$$

is simply the ratio of mass flow rate times specific heat capacity for the two streams. These can be considered as flow stream thermal-capacity rates, i.e., the energy storage rate in the stream per unit of temperature change. [Ref. 11: pp. 14-26]

The attempt taken in this thesis to develop a closed form solution has used the basic fundamentals of heat transfer as well as the foregoing definition. A closed form solution for ε was sought for both 1-5 exchangers with one having three out of five tube passes in parallel flow and the other having three out of five tube passes in counterflow. The analytical approach taken, and demonstrated in this section, is for three out of five tube passes in counterflow.

B. ANALYTICAL DEVELOPMENT

The derivation for the effectiveness, ε, of the 1-5 exchanger as a function of the capacity rate ratio, R, and number of transfer units, N_{tu}, depends on several assumptions.

(1) The overall coefficient of heat transfer, U, does not vary within the exchanger.

(2) The specific heat of both hot side and cold side fluids does not vary.

(3) Each fluid is thoroughly mixed, that is, the temperature of both hot and cold side fluids is uniform over any cross section.

(4) Steady flow conditions are maintained.

(5) Heat losses to or from the environment are negligible.

(6) No change of phase takes place; all heat transferred is sensible heat.

(7) There is equal heat transfer surface in each pass.

The configuration is shown in Figure 3.1 where the five tube passes are designated with subscripts a, b, c, d and e. The temperature of the hot (shell side) fluid is indicated by upper case letters. For the cold (tube side) fluid, lower case letters are used. The subscript 1 always refers to the fluid inlet and the subscript 2 always refers to the fluid outlet.

With W_h and C_{ph} designating mass flow (kg/sec) and specific heat (Joules/kg$^-$°K) of hot fluid entering at T_1 and leaving at T_2 we define a capacity rate for the hot side

$$C_h = W_h C_{ph}$$

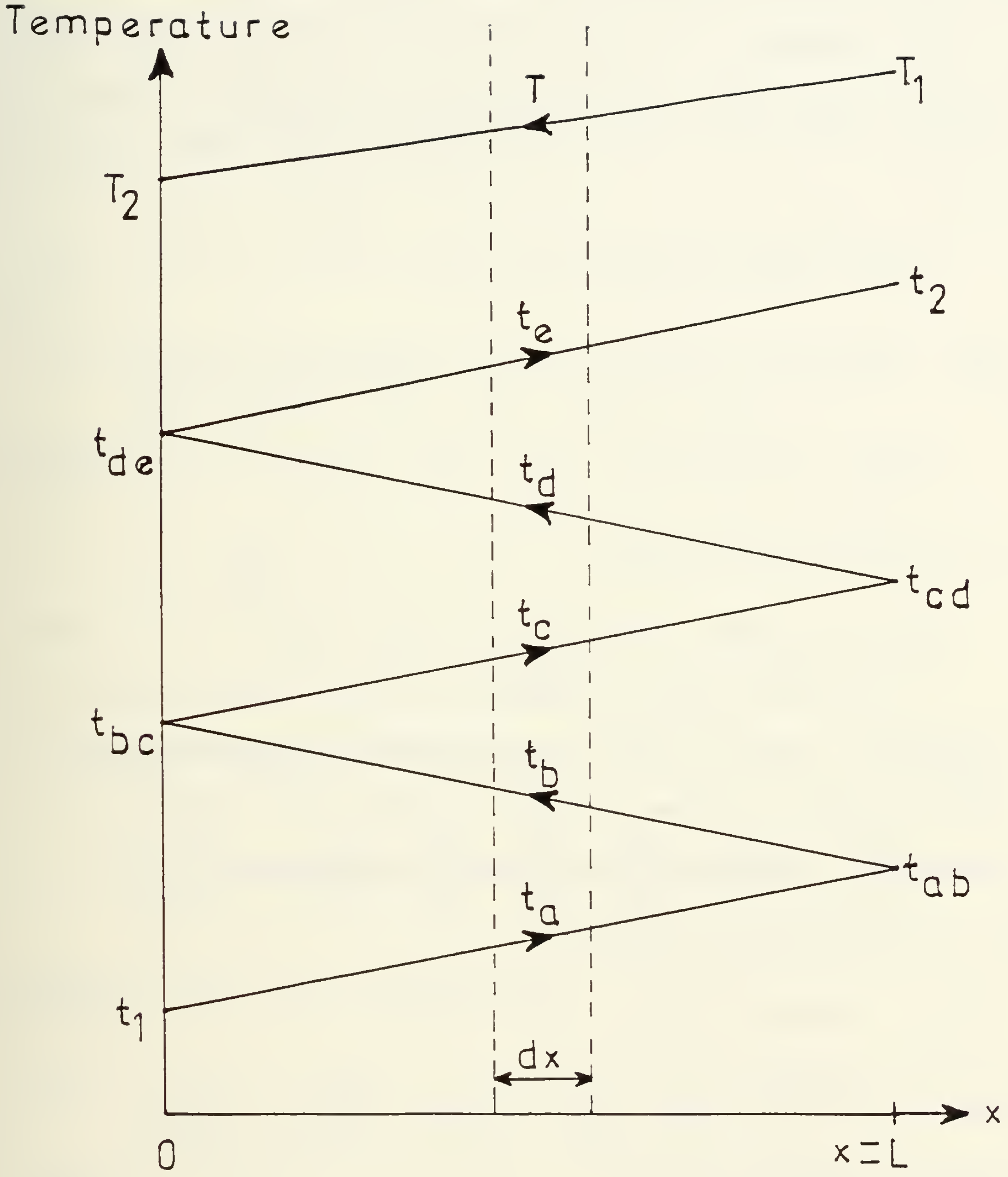

Figure 3.1 Three Counter Two Parallel Pass Configuration for Development of the Sought After Effectiveness Relationship

In similar fashion for the cold side (with W_c and C_{pc}) entering at t_1 and leaving at t_2, we have

$$C_c = W_c C_{pc}$$

We then obtain an energy balance for the entire exchanger

$$C_h(T_1 - T_2) = C_c(t_2 - t_1) \tag{3.1}$$

Over the right hand side of the exchanger (Figure 3.1)

$$C_h(T_1 - T) = C_c(t_2 - t_e + t_d - t_c + t_b - t_a) \tag{3.2}$$

and a differentiation gives

$$C_h dT = C_c(dt_e - dt_d + dt_c - dt_b + dt_a) \tag{3.3}$$

Across dx, with a (m^2/m), the surface per running meter of length of pass so that $A = 5aL$ is the total surface in the exchanger, we may write the heat transferred to the element dx in each cold pass.

$$C_c dt_a = \ \ Uadx(T - t_a) \tag{3.4a}$$

$$C_c dt_b = -Uadx(T - t_b) \tag{3.4b}$$

$$C_c dt_c = \ \ Uadx(T - t_c) \tag{3.4c}$$

$$C_c dt_d = -Uadx(T - t_d) \tag{3.4d}$$

$$C_d dt_e = \ \ Uadx(T - t_e) \tag{3.4e}$$

Here it should be observed that due cognizance has been taken of the direction of the flow in each cold fluid pass with respect to the positive sense of the length coordinate, x, and U is the overall heat transfer coefficient (W/m^2-°C).

With eqs. (3.4) in eq. (3.3)

$$C_h dT = Ua(5T - t_a - t_b - t_c - t_d - t_e)dx$$

or

$$\frac{dT}{dx} = n_h(5T - t_a - t_b - t_c - t_d - t_e) \tag{3.5}$$

where

$$n_h = \frac{Ua}{C_h}$$

is a sort of N_{tu} per unit length for the hot side.

Now differentiate eq. (3.5)

$$\frac{d^2T}{dx^2} = n_h\left(5\frac{dT}{dx} - \frac{dt_a}{dx} - \frac{dt_b}{dx} - \frac{dt_c}{dx} - \frac{dt_d}{dx} - \frac{dt_e}{dx}\right)$$

and with eqs. (3.4) substituted

$$\frac{d^2T}{dx^2} = 5n_h\frac{dT}{dx} - n_c n_h(T - t_a + t_b - t_c + t_d - t_e) \tag{3.6}$$

where

$$n_c = \frac{Ua}{C_c}$$

where again the resemblence of n_c to N_{tu} can be noted.

From eq. (3.2) we obtain

$$\frac{C_h}{C_c}(T_1 - T) - t_2 = t_b - t_a - t_c + t_d - t_e \qquad (3.7)$$

and with eq. (3.7) put into eq. (3.6)

$$\frac{d^2T}{dx^2} - 5n_h \frac{dT}{dx} = - n_c n_h \left[T + \frac{C_h}{C_c}(T_1 - T) - t_2\right]$$

or

$$\frac{d^2T}{dx^2} - 5n_h \frac{dT}{dx} = -n_c n_h \frac{C_h}{C_c}\left[(R_c - 1)T + T_1 - R_c t_2\right] \qquad (3.8)$$

where

$$R_c = C_c/C_h$$

is the capacity rate ratio for the cold side.

Notice that

$$n_c n_h \frac{C_h}{C_c} = \frac{Ua}{C_c} \cdot \frac{Ua}{C_h} \cdot \frac{C_h}{C_c} = \left(\frac{Ua}{C_c}\right)^2 = m$$

and

$$R_h = \frac{1}{R_c} = \frac{C_h}{C_c}$$

a capacity rate ratio for the hot side. Then, algebraic adjustment provides

$$\frac{d^2T}{dx^2} - 5n_h \frac{dT}{dx} + m\left(\frac{1 - R_h}{R_h}\right)T = m\left(\frac{t_2}{R_h} - T_1\right) \tag{3.9}$$

which is a linear, non-homogeneous, second order differential equation with constant coefficients having a complementary function

$$T_c = C_1 e^{\alpha_1 x} + C_2 e^{\alpha_2 x} \tag{3.10}$$

where C_1 and C_2 are arbitrary constants and where

$$\alpha_1, \alpha_2 = \frac{5n_h}{2} \pm \frac{1}{2}\left[25n_h^2 - 4m\left(\frac{1 - R_h}{R_h}\right)\right]^{1/2}$$

$$= \frac{5n_h}{2} \pm \frac{n_h}{2}\left[25 - \frac{4m}{n_h^2}\left(\frac{1 - R_h}{R_h}\right)\right]^{1/2}$$

But

$$\frac{m}{n_h^2} = \frac{(Ua)^2}{(C_c)^2} \cdot \frac{(C_h)^2}{(Ua)^2} = \left(\frac{C_h}{C_c}\right)^2 = R_h^2 = \frac{1}{R_c^2}$$

so that

$$\alpha_1, \alpha_2 = \frac{5n_h}{2} \pm \frac{n_h}{2}\left[25 - 4R_h^2\left(\frac{1 - R_h}{R_h}\right)\right]^{1/2}$$

or

$$\alpha_1, \alpha_2 = \frac{n_h}{2}(5 \pm \lambda) \tag{3.11}$$

where

$$\lambda = [25 - 4R_h(1 - R_h)]^{1/2} \tag{3.12}$$

Designate the particular integral as T_{pi} and by the method of undetermined coefficients let $T_{pi} = P$ so that in eq. (9)

$$m(\frac{1 - R_h}{R_h})\ P = m(\frac{t_2}{R_h} - T_1)$$

This makes

$$T_{pi} = P = [\ \frac{t_2}{R_h} - T_1]\ [\frac{R_h}{1 - R_h}]$$

so that

$$T_{pi} = \frac{t_2 - R_h T_1}{1 - R_h} \tag{3.13}$$

The general solution to eq. (3.9) is the sum of eqs. (3.10) and (3.13)

$$T(x) = C_1 e^{\alpha_1 x} + C_2 e^{\alpha_2 x} + \frac{t_2 - R_h T_1}{1 - R_h} \tag{3.14}$$

where the arbitary constants, C_1 and C_2 are evaluated from conditions at $x = 0$ and $x = L$. At $x = 0$, $T(x = 0) = T_2$ and at $x = L$, $T(x = L) = T_1$. When these are inserted, in turn, into eq. (3.14), one obtains a pair of linear algebraic equations in the unknowns C_1 and C_2

$$T_2 = C_1 + C_2 + T_{pi}$$

$$T_1 = C_1 e^{\alpha_1 L} + C_2 e^{\alpha_2 L} + T_{pi}$$

where T_{pi} is given by eq. (3.13).

It is only a matter of algebra to show that

$$C_1 = \frac{(T_1 - T_{pi}) - (T_2 - T_{pi})e^{\alpha_2 L}}{e^{\alpha_1 L} - e^{\alpha_2 L}} \tag{3.15a}$$

and

$$C_2 = \frac{(T_2 - T_{pi})e^{\alpha_1 L} - (T_1 - T_{pi})}{e^{\alpha_1 L} - e^{\alpha_2 L}} \tag{3.15b}$$

It is easy to see from eq. (3.1) that

$$R_h = \frac{C_h}{C_c} = \frac{(t_2 - t_1)}{(T_1 - T_2)}$$

so that

$$t_2 = t_1 + R_h(T_1 - T_2)$$

Use of this in eq. (3.13) shows that

$$T_{pi} = \frac{t_1 + R_h(T_1 - T_2) - R_h T_1}{1 - R_h}$$

or

$$T_{pi} = \frac{t_1 - R_h T_2}{1 - R_h} \qquad (3.16)$$

indicating two alternative forms for T_{pi} given by eqs. (3.13) and (3.16).

Insertion of eqs. (3.13) and (3.16) in eqs. (3.15) for C_1 and C_2 will yield after some algebra

$$C_1 = \frac{\left(\frac{T_1 - t_2}{1 - R_h}\right) - \left(\frac{T_2 - t_1}{1 - R_h}\right) e^{\alpha_2 L}}{e^{\alpha_1 L} - e^{\alpha_2 L}} \qquad (3.17a)$$

and

$$C_2 = \frac{\left(\frac{T_2 - t_1}{1 - R_h}\right) e^{\alpha_1 L} - \left(\frac{T_1 - t_2}{1 - R_h}\right)}{e^{\alpha_1 L} - e^{\alpha_2 L}} \qquad (3.17b)$$

Equation (3.14) is an expression for the hot side temperature at any location in the exchanger in terms of the extreme temperatures, t_1, t_2, T_1 and T_2.

Next take eq. (3.5) and set it equal to the derivative of eq. (3.14) noting that C_1, C_2 and T_{pi} are all known constants.

$$\frac{dT}{dx} = n_h(5T - t_a - t_b - t_c - t_d - t_e) = \alpha_1 C_1 e^{\alpha_1 x} + \alpha_2 C_2 e^{\alpha_2 x} \qquad (3.18)$$

At $x = 0$, where $T = T_2$, $t_a = t_1$, $t_b = t_c = t_{bc}$, $t_d = t_e = t_{de}$

$$\frac{dT}{dx} = n_h(5T_2 - t_1 - 2t_{bc} - 2t_{de}) = \alpha_1 C_1 + \alpha_2 C_2 \tag{3.19}$$

and if we subtract eq. (3.4a) from eq. (3.4c) we obtain

$$\frac{dt_a - dt_c}{t_a - t_c} = -\frac{Ua}{C_c}\,dx = -n_c dx$$

which can be integrated using C_3 as the constant of integration

$$t_a - t_c = C_3 e^{-n_c x}$$

At $x = 0$ where $t_a = t_1$ and $t_c = t_{bc}$

$$t_1 - t_{bc} = C_3$$

or

$$t_{bc} = t_1 - C_3$$

In addition at $x = L$, $t_a = t_{ab}$ and $t_c = t_{cd}$ so that

$$t_{ab} - t_{cd} = C_3 e^{-n_c L}$$

or

$$C_3 = \frac{t_{ab} - t_{cd}}{e^{-n_c L}} = (t_{ab} - t_{cd})e^{n_c L}$$

This gives a relationship between t_{ab} and t_{bc}

$$t_{bc} = t_1 - (t_{ab} - t_{cd})e^{N_c} \tag{3.20}$$

where $N_c = n_cL$ can be considered as the total number of transfer units for the cold side.

The same procedure may be employed to find t_{de}. Subtract eq. (3.4c) from eq. (3.4e)

$$\frac{dt_c - dt_e}{t_c - t_e} = -\frac{Ua}{C_c}\,dx = -n_c dx$$

and by integration

$$t_c - t_e = C_4 e^{-n_c x}$$

At $x = 0$ where $t_c = t_{bc}$ and $t_e = t_{de}$

$$t_{bc} - t_{de} = C_4$$

or

$$t_{de} = t_{bc} - C_4$$

At $x = L$ where $t_c = t_{cd}$ and $t_e = t_2$

$$t_{cd} - t_2 = C_4 e^{-n_c L}$$

and again with $n_cL = N_c$

$$C_4 = (t_{cd} - t_2)e^{N_c}$$

and

$$t_{de} = t_{bc} - (t_{cd} - t_2)e^{N_c} \qquad (3.21)$$

As before substract eq. (3.4b) from eq. (3.4d)

$$\frac{dt_b - dt_d}{t_b - t_d} = \frac{Ua}{C_c}\,dx = n_c dx$$

and by integrating find that

$$t_b - t_d = C_5 e^{n_c x}$$

at $x = 0$, $t_b = t_{bc}$ and $t_d = t_{de}$ so that

$$t_{bc} - t_{de} = C_5$$

and at $x = L$, $t_b = t_{ab}$ and $t_d = t_{cd}$. This shows that

$$t_{ab} - t_{cd} = C_5 e^{n_c L}$$

and

$$t_{cd} = t_{ab} - (t_{bc} - t_{de})e^{N_c} \qquad (3.22)$$

where $N_c = n_c L$.

Return to eq. (3.18) and look at the conditions at $x = L$ where $t_a = t_b = t_{ab}$, $t_c = t_d = t_{cd}$, $t_e = t_2$ and $T = T_1$. These conditions in eq. (3.18) give

$$n_h(5T_1 - 2t_{ab} - 2t_{cd} - t_2) = \alpha_1 C_1 e^{\alpha_1 L} + \alpha_2 C_2 e^{\alpha_2 L} \qquad (3.23)$$

where again we remember that C_1 and C_2 are known constants.

Now with four equations, eq. (3.20), eq. (3.21), eq. (3.22), and eq. (3.23) and four unknown variables we can solve for the four variables, t_{ab}, t_{bc}, t_{cd}, and t_e. Cramer's rule will be used to solve for two of them; t_{bc} and t_{de}.

First arrange eqs. (3.20) to (3.23) in standard form with the unknowns appearing in columns.

$$t_{ab} + t_{bc}e^{-N_c} - t_{cd} = t_1 e^{-N_c} \tag{3.20}$$

$$- t_{bc} + t_{cd}e^{N_c} + t_{de} = t_2 e^{N_c} \tag{3.21}$$

$$t_{ab} - t_{bc} + t_{cd} + t_{de} = 0 \tag{3.22}$$

$$t_{ab} + t_{cd} = \beta \tag{3.23}$$

where $\beta = 1/2(5T_1 - t_2 - \frac{1}{n_h}(\alpha_1 C_1 e^{\alpha_1 L} + \alpha_2 C_2 e^{\alpha_2 L}))$

If

$$AX = B = \begin{bmatrix} 1 & e^{-N_c} & -1 & 0 \\ 0 & -1 & e^{N_c} & 1 \\ 1 & e^{N_c} & -1 & e^{N_c} \\ 1 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} t_{ab} \\ t_{bc} \\ t_{cd} \\ t_{de} \end{bmatrix} = \begin{bmatrix} t_1 e^{-N_c} \\ t_2 e^{N_c} \\ 0 \\ \beta \end{bmatrix}$$

The solution for t_{bc} and t_{de} is

$$t_{bc} = \frac{\det A_{bc}}{\det A}, \quad t_{de} = \frac{\det A_{de}}{\det A}$$

where, A_{bc} and A_{de} are the matrices obtained by replacing the entries in the second and fourth columns of A by the entries in matrix B.

Compute the determinant, det A of matrix A (the matrix of coefficients)

$$\det A = \begin{vmatrix} 1 & e^{-N_c} & -1 & 0 \\ 0 & -1 & e^{N_c} & 1 \\ 1 & e^{N_c} & -1 & e^{N_c} \\ 1 & 0 & 1 & 0 \end{vmatrix}$$

$$\det A = -(e^{N_c} + 2e^{-N_c})$$

Since det A $\neq 0$, the system should have a unique solution. Continue by computing det A_{bc} and det A_{de}. Thus

$$\det A_{bc} = \begin{vmatrix} 1 & t_1e^{-N_c} & -1 & 0 \\ 0 & t_2e^{N_c} & e^{N_c} & 1 \\ 1 & 0 & -1 & e^{N_c} \\ 1 & \beta & 1 & 0 \end{vmatrix}$$

$$= e^{2N_c}(\beta - 2t_2) - t_1(e^{N_c} + 2e^{-N_c})$$

and

$$\det A_{de} = \begin{vmatrix} 1 & e^{-N_c} & -1 & t_1 e^{-N_c} \\ 0 & -1 & e^{N_c} & t_2 e^{N_c} \\ 1 & -e^{N_c} & -1 & 0 \\ 1 & 0 & 1 & \beta \end{vmatrix}$$

$$= (e^{2N_c} + 1)(\beta - 2t_2) - t_1(e^{N_c} + 2e^{-N_c})$$

Therefore

$$t_{bc} = \frac{\det A_{bc}}{\det A} = \frac{e^{2N_c}(\beta - 2t_2) - t_1(e^{N_c} + 2e^{-N_c})}{-(e^{N_c} + 2e^{-N_c})}$$

or

$$t_{bc} = t_1 - \frac{e^{N_c}(\beta - 2t_2)}{1 + 2e^{-2N_c}} \qquad (3.24)$$

and

$$t_{de} = \frac{\det A_{de}}{\det A} = \frac{(e^{2N_c} + 1)(\beta - 2t_2) - t_1(e^{N_c} + 2e^{-N_c})}{-(e^{N_c} + 2e^{-N_c})}$$

or

$$t_{de} = t_1 - \left(\frac{(e^{2N_c} + 1)(\beta - 2t_2)}{e^{N_c} + 2e^{-N_c}} \right) \qquad (3.25)$$

Then with eq. (3.24) and (3.25) in eq. (3.19) and some algebraic manipulation

$$\alpha_1 C_1 + \alpha_2 C_2 = n_h [5(T_2 - t_1) + 2(\frac{(\beta - 2t_2)(2e^{N_c} + e^{-N_c})}{(1 + 2e^{-2N_c})})] \tag{3.26}$$

Equation (3.26) is similar to the one derived by O'Hare [Ref. 1: eq. (22)]. At this point he continued to attempt a closed form solution for ε as a function of R and N_{tu}. It is to be noted, however, that he was unable to obtain the solution because of the presence of terms deriving from dimensional parameters such as n_h with dimensions m^{-1} [Ref. 1: pp. 46]. Since n_h is present in eq. (3.26), it must be concluded that a further attempt at an analytic solution for the effectiveness of the 1-5 exchanger will suffer a similar fate and be also unachieveable.

This conclusion should be obvious because, when dealing with an equation derived from n equations with n+1 unknowns, one cannot create an additional equation by merely multiplying one of the n equations by a constant. This leads to a linearly dependent set which has no unique solution.

Thus, after observing that O'Hare's attempt failed, this thesis, like O'Hare's, turns to a numerical analysis to achieve the requisite objective.

IV. NUMERICAL AND COMPUTER ANALYSIS

As O'Hare discovered with the 1-3 heat exchanger [Ref. 1], an analytic solution for effectiveness as a function of the capacity rate ratio, R, and N_{tu} was also unattainable for the 1-5 exchanger. Hence, an alternate approach had to be taken and this involved utilizing a finite-difference thermal analyzer combined with a linearizing scheme for the solution of a set of nonlinear algebraic equations [Ref. 14]. This thermal analyzer has been employed to provide solutions for temperature that lead to effectiveness, ε, value for the 1-5:3C and 1-5:3P heat exchangers as a function of R and N_{tu}.

A. THERMAL ANALYZER TVSSI

The computer program TVSSI used in the development of the $\varepsilon = f(R, N_{tu})$ graphs for the 1-5 heat exchanger [Ref. 1: Appendix A], was adapted from TVSS2 listed by Kern and Kraus [Ref. 14: Appendix C]. This adaptation consisted of modifying TVSS2 to compute solutions in the SI system and accept an input file specifically designed for the 1-5 heat exchanger. This program also had to be adapted to the IBM 3033 AP system. The program utilizes the Cholesky decomposition scheme, and because of the linearization of the radiation terms (not used in this study) the program is iterative.

B. DEVELOPMENT OF THE 1-5:3C and 1-5:3P HEAT-EXCHANGER MODELS

As indicated in the foregoing, TVSSI uses an input file to generate a solution for temperatures that will lead to the sought after effectiveness as a function of R and N_{tu}. This input file is created by a program which requires an input of given capacity rates (hot and cold), overall coefficient, surface area and inlet stream temperatures (hot and cold) values.

This required the development of a program which created such a file. O'Hare [Ref. 1: pp. 53-55] first developed a program using the 1-4 heat exchanger that would create an input file for TVSSI. He compared the effectiveness from the computer generated results and a known analytic solution of effectiveness developed by Kraus and Kern [Ref. 8]. Once O'Hare established confidence in his results, he proceeded to develop a program which would create an input file for the 1-3 exchanger. In this study, a similar technique was used to develop a computer program for the 1-5:3C and 1-5:3P heat exchangers. These programs are called NTU53C and NTU53P (see Appendices A and B) and, in them, the following parameters and techniques were utilized.

1. 300 nodes were used.
2. The initial temperature for the computer to begin the iterative process was set at 200°C.
3. An eventual accuracy of .05 between the final and next to last iterations was used.

4. A radiation coefficient convergence factor of 0.66667 between iterations was used. This is required by the program even if radiation is not considered as a node.

5. The maximum number of iterations that the computer was allowed to perform was set at 12.

6. A damping factor of .8 was set as an initial damping based on the number of non-linear terms in all of the node equations.

With the values of C_h, C_c, U, A, T_1 and t_1 set, the input is set and an input file for TVSSI was generated. In this file all node equations and internode conductance values were determined. This program specified the nodes that interact with each other and the methods by which the thermal interaction takes place such as in the forced convection and fluid flow modes.

Both programs make use of the fact that each term in a node equation shows three things. The first is the node that is coupled for heat flow with the node in question. The second is the method of heat flow between the nodes. In this case,forced convection and fluid flow are used. Finally, the node equation shows the magnitude of the internode heat flow. Here all the pieces of information are collected and presented for use by TVSSI as an input file with all items in the proper format.

It is the output values of both NTU53C and NTU53P (a typical result is shown in Appendix C) that are used by the thermal analyzer (as an input file) to determine the temperatures T_2 and t_2 for the specific set of given initial parameters C_h, C_c, R, A, T_1, and t_1.

C. SCOPE OF COMPUTER ANALYSIS

With the development of NTU53C and NTU53P computer programs, the files that they generated could then be used to solve for temperatures which in turn could yield effectiveness values for a given R and N_{tu}. Of course, it must be realized that many computer runs are required to generate enough data to ensure confidence in the results which cover a wide range of capacity rate ratios and N_{tu} values.

To efficiently expedite the computer task, the Multiple Virtual System (MVS) with Job Entry Subsystem and Networking (JES3) was employed. The MVS coupled with JES3 is more commonly referred to as batch processing. Based on trial and error, it was determined that in order to build a solid data base, eleven different values for effectiveness were needed to best represent a particular value of R. Thus, a range of R from R = 0.01 to 1.0 was used. Due to the enormous amount of computer time required to solve a 300 node problem, R increments of 0.05 were used for this study and this thesis reports on this basis. In all, 42 curves for both the 1-5:3C and 1-5:3P heat exchangers were required. With this change in the increments between R values (O'Hare's work showed that an increment in R of 0.01 was not necessary), a linear interpolation was found to be acceptable to less than 0.02 percent error.

TVSSI was slightly modified in accordance with the appropriate guidelines of the job control language (JCL) needed to run on the batch processing system. These modifications were few and were needed only at the beginning and end of TVSSI. The modified version of TVSSI has been called TVCOUNT with changes shown in [Ref. 1: Appendix F]. It was TVCOUNT that was then used to activate TVSSI.

It also became necessary to modify the two input file programs NTU53P and NTU53C so that they needed to be compiled only once. They were then loaded in a library file to be used when called by another program. New programs utilizing the batch system were written that could easily be loaded with the appropriate input data for a specific R value. These, which are referred to as "sister programs," were used to go from the library file to TVSSI and cause TVSSI to be executed eleven times under TVCOUNT covering the desired range of N_{tu} for a specific R value. The modified input files and their associated "sister execution programs are found in Appendices D through G.

A flow chart of how all of the foregoing is accomplished is provided by O'Hare [Ref. 1: Figures 4.4 and 4.5]. It is noted that in these figures, TVSSI is referred to as TVSSIA through TVSSIV. These are the same programs as TVSSI but for bookkeeping purposes by the computer they are labeled A through V.

D. INTERPRETATION OF GRAPHS

Once all the data had been collected and the effectiveness for a given R and N_{tu} values had been computed, plots of effectiveness as a function of N_{tu} for the entire range of R values were plotted for both parallel and counterflow heat exchangers. These plots are shown in Appendices H and I.

It may be observed that for a particular value of R, the plot of effectiveness as a function of N_{tu} indicates that the 1:5-3C exchanger outperforms the 1-4, 1-3:2P and the 1-5:3P exchangers but not the 1-3:2C exchanger. This may be explained by the quite apparent fact that the 1-5:3C exchanger has one more counterflow pass than the 1-4 exchanger. Indeed, this argument can be extended to the fact that the 1-5:3C has one more parallel flow pass than the 1-3:2C. Figure 4.1. displays the graphical picture to support this contention.

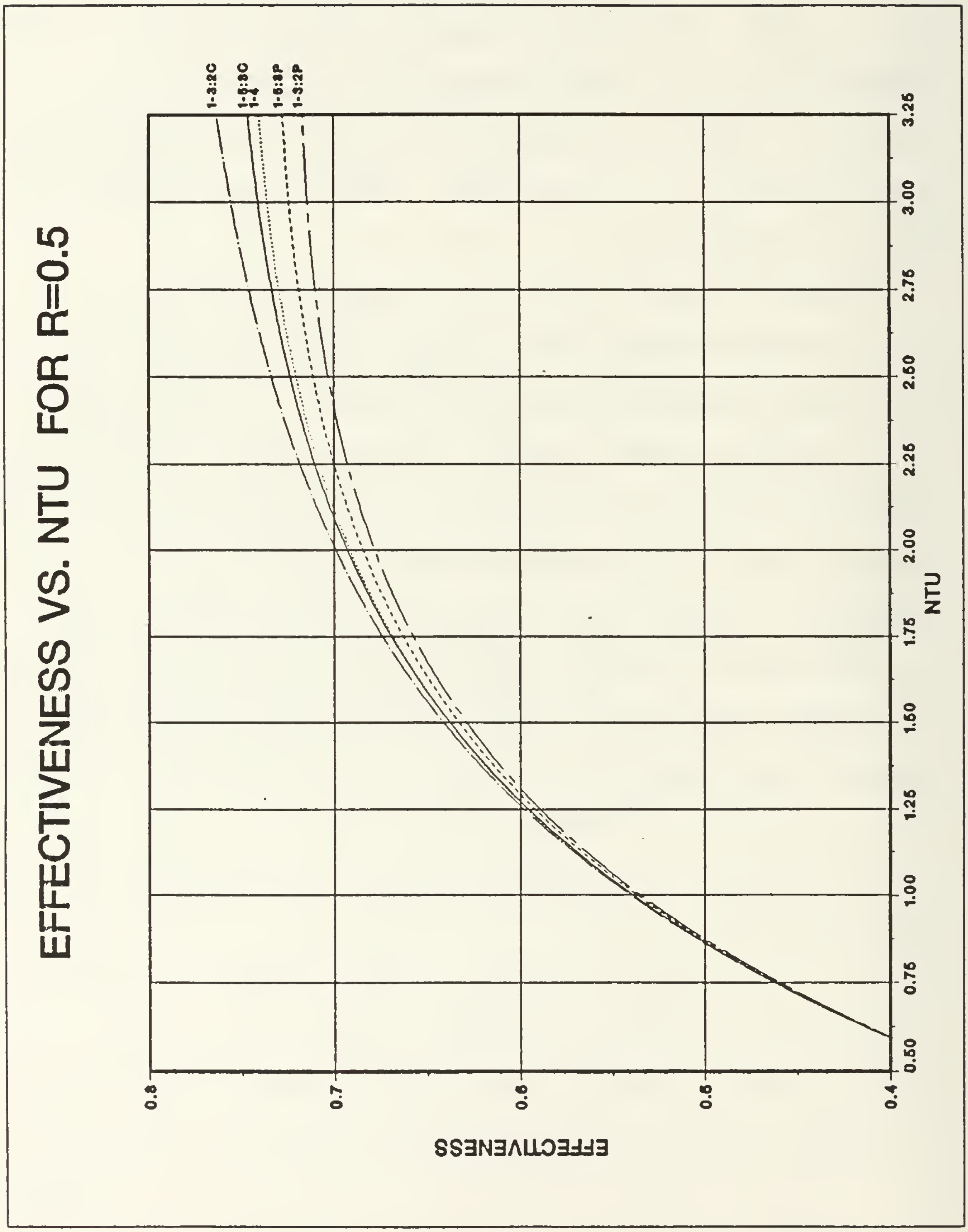

Figure 4.1 Comparison of Analytical (1-4) and Computer Results (1-3:2C and 1-3:2P) to Computer Results (1-5:3C and 1-5:3P) at R = 0.5

V. POLYNOMIAL REGRESSION

A. DEVELOPMENT OF POLYNOMIAL EQUATIONS

The empirical data obtained for the two 1-5 heat exchangers was designed to cover an extensive range of R values varying from 0.01 to 1.0 in increments of 0.05. As discussed earlier in Section IV, the data obtained at a specific value of R is the computer evaluated result of the effectiveness, for an associated N_{tu} value. With this accomplished, it then becomes possible to graph separate curves for each of the different R values as shown in Appendices H and I. Through a polynomial regression technique, as discussed in this section, it is also possible to develop implicit equations for the curves with $\varepsilon = f(N_{tu}, R)$. It is also apparent from an inspection of the graphical representation of the empirical data in Appendices H and I, that the curves conform to a high degree polynomial. However, further analytical investigation is needed to ascertain the exact order of the polynomial terms. This investigation will not only lead to the order of the polynomial, but to the specific equation for each curve.

By use of polynomial regression, the least-squares method can be readily extended to best fit the data to the m^{th}-degree for the polynomial

$$y = A_0 + A_1 x + A_2 x^2 + \dots A_m x^m \tag{5.1}$$

with the error defined by

$$e_i = D_i - y_i = D_i - A_0 - A_1 x - A_2 x_i^2 - \ldots - A_m x_i^m$$

where D_i represents the empirical data value corresponding to x_i, x_i being free of error.

The objective is to minimize the sum of the squares of the residuals, S_r,

$$S_r = \sum_{i=1}^{m} e_i^2 = \sum^{m} (D_i - A_0 - A_1 x_1 - A_2 x_2^2 + \ldots A_m x^m)^2 \quad (5.2)$$

Because at a minimum, the partial derivatives $\partial S_r/\partial A_0$, $\partial S_r/\partial A_1$ ... $\partial S_r/\partial A_m$ vanish, after taking the derivative of S_r with respect to each of the coefficients of the polynomial, it can be seen that

$$\frac{\partial S_r}{\partial A_0} = 0 = -2 \Sigma (D_i - A_0 - A_1 x_i - A_2 x_i^2 - \ldots - A_m x_i^m)$$

$$\frac{\partial S_r}{\partial A_1} = 0 = -2 \Sigma x_i (D_i - A_0 - A_1 x_i - A_2 x_i^2 - \ldots - A_m x_i^m)$$

$$\frac{\partial S_r}{\partial A_2} = 0 = -2 \Sigma x_i^2 (D_i - A_0 - A_1 x_i - A_2 x_i^2 - \ldots - A_m x_i^m)$$

$$\begin{array}{ccc} \vdots & \qquad\qquad & \vdots \end{array}$$

$$\frac{\partial S_r}{\partial A_m} = 0 = -2 \Sigma x_i^m (D_i - A_0 - A_1 x_i - A_2 x_i^2 - \ldots - A_m x_i^m)$$

Then by dividing by -2 and rearranging we obtain

$$\begin{array}{l}
A_0 N + A_1 \Sigma x_i + A_1 x_i^2 + \ldots + A_m \Sigma x_i^m = \Sigma D_i \\
A_0 \Sigma x_i + A_1 \Sigma x_i^2 + A_2 \Sigma x_i^3 + \ldots + A_m \Sigma x_i^{m+1} = \Sigma x_i D_i \\
A_0 \Sigma x_i^2 + A_1 \Sigma x_i^3 + A_2 \Sigma x_i^4 + \ldots + A_m \Sigma x_i^{m+2} = \Sigma x_i^2 D_i \\
\quad \vdots \qquad\qquad \vdots \qquad\qquad \vdots \qquad\qquad\qquad\qquad \vdots \qquad\qquad \vdots \\
A_0 \Sigma x_i^m + A_1 \Sigma x_i^{m+1} + A_2 \Sigma x_i^{m+2} + \ldots + A_m \Sigma x_i^{2m} = \Sigma x_i^m D_i
\end{array}$$

where all summations are from i=1 through n. All of the foregoing m+1 equations are linear and have m+1 unknowns: A_0, A_1, A_2, ... A_m. The coefficients of the unknowns can be calculated directly from the observed data. Thus, the problem of determining a least-squares polynomial of degree m is equivalent to solving a system of m+1 simultaneous linear equations. Putting the equations in matrix form yields

$$\begin{bmatrix}
N & \Sigma x_i & \Sigma x_i^2 & \Sigma x_i^3 & \ldots & \Sigma x_i^m \\
\Sigma x_i & \Sigma x_i^2 & \Sigma x_i^3 & \Sigma x_i^4 & \ldots & \Sigma x_i^{m+1} \\
\Sigma x_i^2 & \Sigma x_i^3 & \Sigma x_i^4 & \Sigma x_i^5 & \ldots & \Sigma x_i^{m+2} \\
\vdots & \vdots & \vdots & \vdots & & \vdots \\
\Sigma x_i^m & \Sigma x_i^{m+1} & \Sigma x_i^{m+2} & \Sigma x_i^{m+3} & \ldots & \Sigma x_i^{2m}
\end{bmatrix} A = \begin{bmatrix}
\Sigma D_i \\
\Sigma x_i D_i \\
\Sigma x_i^2 D_i \\
\vdots \\
\Sigma x_i^m D_i
\end{bmatrix}$$

[Ref. 15: pp. 302-309 and Ref. 16: 468-474].

From this point on, one finds that it is best to use a computer to assist in solving the simultaneous equations and this will also help alleviate any ill-conditioning that may otherwise occur. An existing "curvefit" program available through NON-IMSL [Ref. 16] was used although some modifications were made to the original program to best accommodate the goals of this work.

To determine the order of polynomial that should eventually be used, one increases the degree of the approximating polynomial as long as there is a statistically significant decrease in the variance σ^2, which is computed by

$$\sigma^2 = \frac{\Sigma e_i^2}{N - m - 1} \tag{5.3}$$

In otherwords, the selection of the optimum degree polynomial is contingent upon a decreasing variance and once the variance begins to increase, the degree of the polynomial becomes too high. For all cases, it was found that the $\varepsilon - N_{tu}$ developed curves are of the 5th order.

As shown in Figures 5.1 and 5.2 the computed values of effectiveness vs. N_{tu} for R = 0.1, 0.5 and 1.0 for both flow arrangements, (1-5:3P) and (1-5:3C), have been graphed and fitted by a 5th degree polynomial. Because all computed values for effectiveness follow a predictable trend, only a sample of the data covering the whole range of

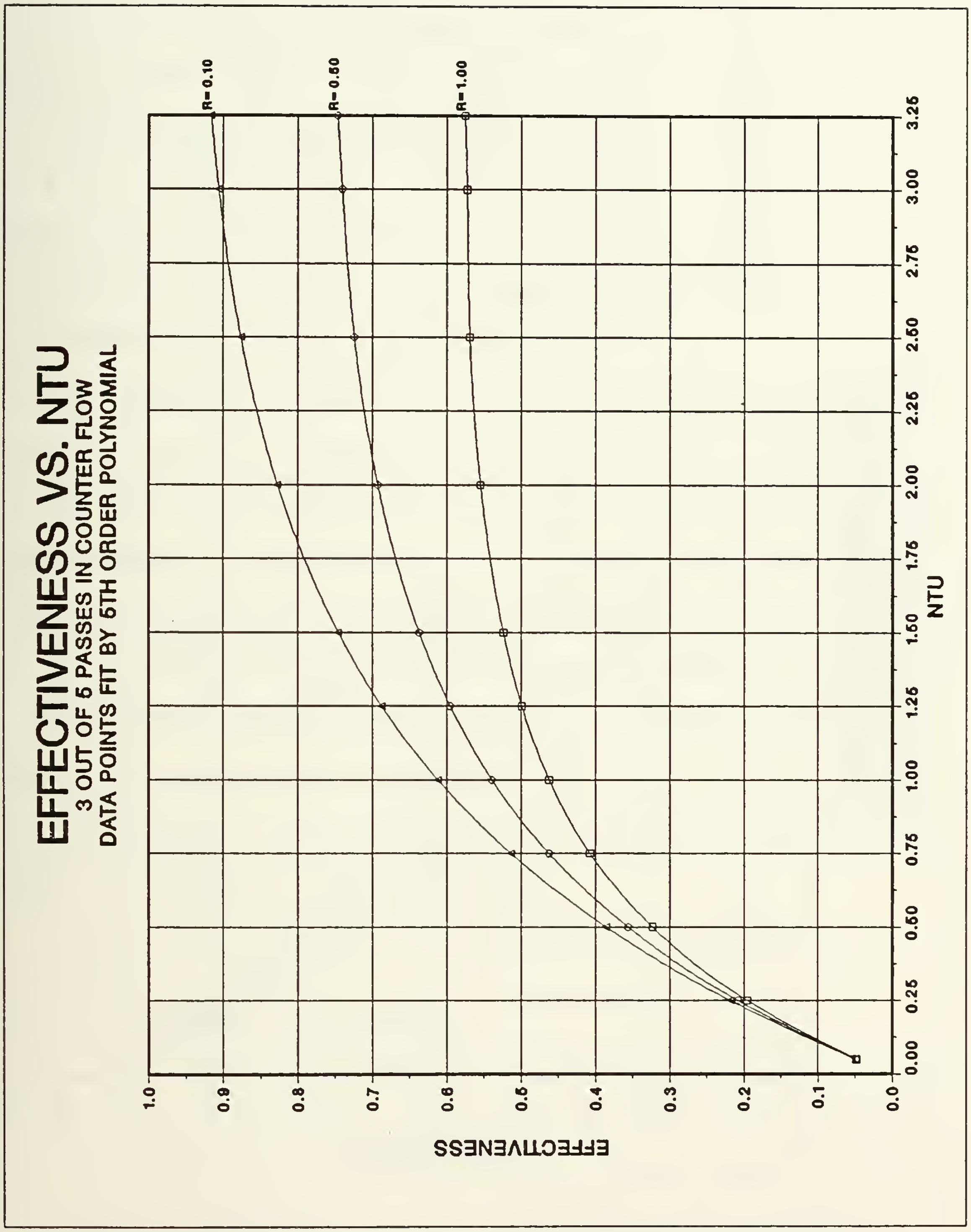

Figure 5.1 1-5:3C Data Fit by a 5th Order Polynomial

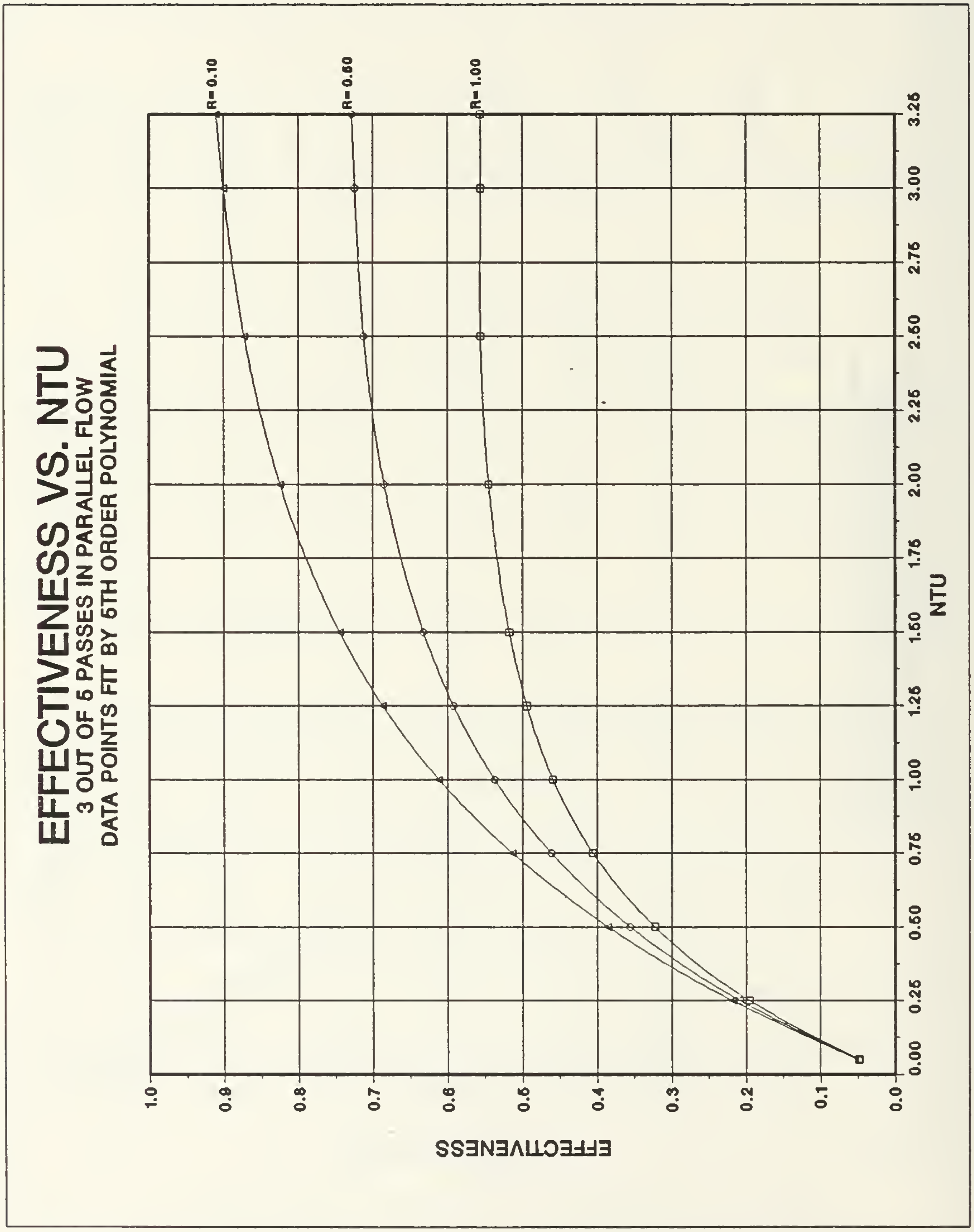

Figure 5.2 1-5:3P Data Fit by a 5th Order Polynomial

values of R have been shown. It is clear that the graphical interpretation strongly backs what is known numerically from the polynomial regression technique. Thus, the relationship for $\varepsilon = f(N_{tu}, R)$ can be found explicitly from

$$\varepsilon = A_5 N_{tu}^5 + A_4 N_{tu}^4 + A_3 N_{tu}^3 + A_2 N_{tu}^2 + A_1 N_{tu} + A_0 \quad (5.4)$$

where the coefficients A_5, A_4, A_3, A_2, A_1, and A_0 relating to a specific value of R are found in Tables 1 and 2 for the (1-5:3C) and (1-5:3P) configurations.

In order to better utilize the coefficients found in Tables 1 and 2, an interactive program has been developed using equation (5.4) and the 1-5 heat exchanger polynomial approximation coefficients. This program was written for the IBM 3033 using the FORTRAN VS compiler (See Appendix J). By entering the values for NTU, R and the type of heat exchanger the effectiveness values are readily obtained directly or by linear interpolation.

An example of how to work with the effectiveness by using the Equation (5.4) follows.

B. NUMERICAL EXAMPLE

1. Statement of Problem

A 1-5:3C heat exchanger uses sea water (Cp = 4.045 kJ/kg-°C) at an initial temperature of 20°C to cool engine oil, flowing at 200 kg/s from 60°C to 40°C. The sea water flow rate is 80 kg/s. If the overall heat transfer coefficient is 40 W/m^2°C, how much surface area is required in the exchanger?

2. Solution

At an average oil temperature of 50°C, the specific heat is C_p = 2.004 kJ/kg-°C and an energy balance gives

$$(WC_p\Delta T)_{\text{sea water}} = (WC_p\Delta T)_{\text{engine oil}}$$

$$(80\text{ kg/s})(4.045\text{ kJ/kg-°C})(t_2 - 20)\text{°C}$$

$$= (200\text{ kg/s})(2.004\text{ kJ/kg-°C})(60 - 40)\text{°C}$$

$$t_2 = 20 + \frac{200(2.004)(20)}{10(4.045)} = 44.8\text{°C}$$

The number of transfer units is given by N_{tu} = UA/C_{min}. To determine C_{min} we must compare the products of mass-flow rate and specific heat for sea water and oil.

$$(WC_p)_{sw} = (80\text{ kg/s})(4.045\text{ kJ/kg-°C})$$

$$= 323.6\text{ kW/°C}$$

$$(WC_p)_{oil} = (200\text{ kg/s})(2.004\text{ kJ/kg-°C})$$

$$= 400.8\text{ kW/°C}$$

Therefore C_{min} = 323.6 kW/°C.

The effectiveness is a function of R and N_{tu}. The capacity rate ratio, R, is

$$R = \frac{C_{min}}{C_{max}} = \frac{323.6}{400.8} = 0.8074$$

Because $C_c = C_{min}$, the required effectiveness is

$$\frac{t_2 - t_1}{T_1 - t_1} = \frac{44.8 - 20}{60 - 20} = 0.6193$$

We must now go to Table 1 for the 1-5:3C arrangement with $R = 0.8$ and $R = 0.85$ and find the coefficients

$R = 0.8$	$R = 0.85$
$A_0 = 3.63419 \times 10^{-3}$	$A_0 = 3.82659 \times 10^{-3}$
$A_1 = 0.93822$	$A_1 = 0.93407$
$A_2 = -0.67053$	$A_2 = -0.67841$
$A_3 = 0.27531$	$A_3 = 0.28109$
$A_4 = -0.60558 \times 10^{-1}$	$A_4 = -0.62088 \times 10^{-1}$
$A_5 = 0.54393 \times 10^{-2}$	$A_5 = 0.55850 \times 10^{-2}$

A trial and error solution (assuming values of N_{tu} and by interpolating) gives a result

$$N_{tu} = 2.352$$

for which

$$A = N_{tu}\, C_{min}/U$$

$$A = 19.03 \text{ m}^2$$

3. Observations

The primary observation made here is that by using the 5th order polynomial equation with the appropriate coefficients found in Table 1 or 2, an accurate value for effectiveness can be found thus allowing one to solve for a wide variety of heat exchanger parameters.

By comparing the value for effectiveness just computed with that of the 1-5:3P, 1-3:2P [Ref. 1] and 1-4 to 1-12 even pass exchangers [Ref. 8], the 1-5:3C has a higher effectiveness than all (see Table 3).

From this observation it is evident that the 1-5:3C exchanger outperforms not only the 1-4 to 1-12 even pass arrangements but also its counterpart, the 1-5:3P exchanger for these given values by at least 0.614%.

TABLE 1

1-5:3C COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.01	0.57463	0.99093	-0.47938	0.13879	-0.23075	0.16779
.05	0.79100	0.98831	-0.49081	0.14654	-0.25068	0.18647
.10	1.08210	0.98622	-0.50716	0.15773	-0.28045	0.21547
.15	1.25311	0.98238	-0.52072	0.16779	-0.30947	0.24635
.20	1.42506	0.97866	-0.53259	0.17545	-0.32772	0.26203
.25	1.33145	0.97670	-0.54709	0.18482	-0.35072	0.28213
.30	1.51081	0.97405	-0.56166	0.19541	-0.38055	0.31252
.35	1.74198	0.97091	-0.57527	0.20542	-0.40903	0.34186
.40	1.84400	0.96879	-0.59067	0.21718	-0.44469	0.38122
.45	2.13420	0.96459	-0.60073	0.22416	-0.46295	0.39861
.50	2.30808	0.96140	-0.61293	0.23322	-0.48933	0.42669
.55	2.57953	0.95713	-0.62234	0.23996	-0.50762	0.44499
.60	2.74755	0.95336	-0.63230	0.24706	-0.52681	0.46391
.65	3.05249	0.94903	-0.64122	0.25366	-0.54530	0.48293
.70	3.19573	0.94580	-0.65184	0.26137	-0.56656	0.50422
.75	3.39253	0.94193	-0.66116	0.26833	-0.58615	0.52431
.80	3.63419	0.93822	-0.67053	0.27531	-0.60558	0.54393
.85	3.82659	0.93407	-0.67841	0.28109	-0.62088	0.55850
.90	3.98221	0.93052	-0.68748	0.28799	-0.64051	0.57874

TABLE 1 (cont'd)

1-5:3C COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.95	4.26945	0.92609	-0.69496	0.29381	-0.65653	0.59442
1.0	4.40670	0.92259	-0.70365	0.30046	-0.67514	0.61309

TABLE 2

1-5:3P COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.01	0.52233	0.99152	-0.48055	0.13953	-0.23283	0.16989
.05	0.81101	0.98817	-0.49062	0.14604	-0.24883	0.18421
.10	1.10555	0.98599	-0.50718	0.15721	-0.27865	0.21337
.15	1.14518	0.98309	-0.52246	0.16811	-0.30968	0.24604
.20	1.25862	0.97914	-0.53367	0.17485	-0.32443	0.25741
.25	1.29233	0.97701	-0.54864	0.18464	-0.34902	0.27929
.30	1.43113	0.97558	-0.56664	0.19807	-0.38885	0.32195
.35	1.69053	0.97206	-0.57958	0.20740	-0.41500	0.34854
.40	1.76316	0.96998	-0.59515	0.21912	-0.45054	0.38805
.45	2.10897	0.96559	-0.60498	0.22583	-0.46805	0.40487
.50	2.30583	0.96163	-0.61546	0.23327	-0.48857	0.42594
.55	2.59035	0.95726	-0.62480	0.23985	-0.50625	0.44354
.60	2.77374	0.95331	-0.63434	0.24638	-0.52303	0.45925
.65	2.91148	0.95025	-0.64574	0.25460	-0.54593	0.48260
.70	3.22294	0.94575	-0.65400	0.26039	-0.56053	0.49575
.75	3.38351	0.94231	-0.66425	0.26788	-0.58110	0.51617
.80	3.60234	0.93852	-0.67369	0.27495	-0.60079	0.53606
.85	3.82374	0.93414	-0.68130	0.28045	-0.61487	0.54884
.90	3.91805	0.93133	-0.69231	0.28893	-0.63953	0.57472

TABLE 2 (cont'd)

1-5:3P COEFFICIENTS

R	A_0 x10^3	A_1	A_2	A_3	A_4 x10^1	A_5 x10^2
.95	4.18844	0.92723	-0.70050	0.29533	-0.65754	0.59298
1.0	4.35186	0.92385	-0.71022	0.30310	-0.68037	0.61690

TABLE 3

COMPARISON OF EXAMPLE RESULTS FOR Ntu = 2.352 AND R = 0.8074

NUMBER OF TUBE PASSES	EFFECTIVENESS
3:2P	.5990
4	.6155
5:3C	.6193
5:3P	.6067
6	.6145
8	.6141
10	.6140
12	.6139

VI. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The objective of this thesis was to provide effectiveness values for the 1-5:3C and 1-5:3P heat exchangers. Although these objectives were analytically unachievable, they became attainable through a fifth order polynomial approximation from numerical data. The data obtained covers a wide range of capacity rate ratios for values of N_{tu} from 0 to 3.25. With the knowledge of the 1-5 heat exchanger type (1-5:3C or 1-5:3P), size (surface area), the overall heat transfer coefficient and the fluid flow rates, the N_{tu} and capacity rate ratios may be computed and the exchanger effectiveness may then be determined from the appropriate coefficient tables or the effectiveness program provided.

It is indeed obvious that the effectiveness parameter is most useful in performance estimation and analysis for all types of heat exchangers. The work done here pertaining to the 1-5 exchanger is very helpful in constructing a comprehensive picture of the performance attainable for the 1-2 consecutively through the 1-6 for a diverse range of known conditions.

B. RECOMMENDATIONS

With proof in this thesis, and that of O'Hare [Ref. 1], that an odd number of tube passes per shell has a slightly better effectiveness when more than half the tube fluid flows in counterflow to the shell fluid, it is recommended that a feasibility study be conducted with regard to the structural and thermal problems that may arise in manufacturing and designing such heat exchangers.

APPENDIX A

NTU53C COMPUTER GENERATED INPUT ANALYZER PROGRAM

```
C         THIS IS PROGRAM NTU53C
C
C
C         IT GENERATES AN INPUT FILE FOR THERMAL ANALYZER TO OBTAIN
C         1-5 EFFECTIVENESS-NTU RELATIONSHIP THAT IS NOT AVAILABLE IN
C         OPEN LITERATURE (3C MEANING two PARALLEL PASS AND THREE
C         COUNTERFLOW PASSES).
C
      DIMENSION COEF(300,6),KCON(300,6),L1(8),L2(3),L3(6),SET2(2),FL4(4)
C
      CHARACTER *79 TITLE
      CHARACTER *12 FNAME
C
      DATA IOT,IN,IPR,IWR/6,5,4,8/
C
C     OPEN PRINTER OUTPUT FILE
C
      OPEN(IPR,FILE='PRN',STATUS='NEW',FORM='FORMATTED',IOSTAT=ICK)
      IF(ICK.NE.0) WRITE(IOT,920)
  920 FORMAT(' Trouble opening printer output file' )
C
      WRITE(IOT,917)
  917 FORMAT(/' Input the title of this study - 79 columns only. ',
     & ' This title will appear',/,' at the top of every printed page',
     & ' of output:')
      READ(IN,918) TITLE
  918 FORMAT(A79)
C
      WRITE(IOT,901)
  901 FORMAT(/' INPUT HOT SIDE CAPACITY RATE:')
      READ(IN,902) CHOT
  902 FORMAT(BN,F10.0)
C
      WRITE(IOT,903)
  903 FORMAT(/' INPUT COLD SIDE CAPACITY RATE:')
      READ(IN,902) CCLD
C
      WRITE(IOT,904)
  904 FORMAT(/' INPUT OVERALL HEAT TRANSFER COEFFICIENT:')
      READ(IN,902) U
C
      WRITE(IOT,905)
  905 FORMAT(/' INPUT TOTAL HEAT TRANSFER SURFACE:')
      READ(IN,902) SURFTO
C
      WRITE(IOT,906)
  906 FORMAT(/' INPUT HOT SIDE INLET TEMPERATURE:')
      READ(IN,902) THOTIN
C
      WRITE(IOT,907)
  907 FORMAT(/' INPUT COLD SIDE INLET TEMPERATURE:')
      READ(IN,902) TCLDIN
C
      VALK1 = CHOT
      VALK2 = CCLD
      VALK3 = U*SURFTO/250.
      TINIT = 125.
C
C             FRONT END
C
      L1(1) = 300
      L1(2) = 2
      DO 10  I=3,8
   10 L1(I) = 0
C
      DO 20  I=1,3
   20 L2(I) = 0
C
      L3(1) = 300
      L3(2) = 50
      L3(3) = 6
      L3(4) = 2
      L3(5) = 4
      L3(6) = 6
C
      FL4(1) = .05
      FL4(2) = .66667
      FL4(3) = .8
      FL4(4) = TINIT
      L4 = 12
C
C    CONSTANT TEMPERATURES
C
      SET2(1) = THOTIN
      SET2(2) = TCLDIN
C
C    READY FOR INPUT SET 4
C
C    NODE 1
C
      KCON(1,1) = 1004
      KCON(1,2) = 1014
      KCON(1,3) = 2004
      KCON(1,4) = 2014
      KCON(1,5) = 3004
      KCON(1,6) = 3015
      COEF(1,6) = VALK1
      DO 50 I = 1,5
   50 COEF(1,I) = VALK3
C
C    NODES 2 TO 50
C
      DO 75 I = 2,50
      J = 101 - I
      K = 100 + I
      L = 201 - I
      N = I - 1
      M=200+I
      MM=301-I
      KCON(I,1) = 10*N + 5
      KCON(I,2) = 10*J + 4
      KCON(I,3) = 10*K + 4
      KCON(I,4) = 10*L + 4
      KCON(I,5) = 10*M + 4
      KCON(I,6) = 10*MM + 4
      COEF(I,1) = VALK1
      DO 80  II = 2,4
      COEF(I,II) = VALK3
   80 CONTINUE
   75 CONTINUE
C
C   NODE 51
C
      KCON(51,1) = 3025
      KCON(51,2) = 504
      COEF(51,1) = VALK2
      COEF(51,2) = VALK3
C
C    NODES 52 TO 300
C
      DO 120  I = 52,300
      K = I - 1
      IF(I.GT.100) GO TO 122
      L = I - 50
      M = 2*L - 1
      J = I - M
      GO TO 135
  122 IF(I.GT.150) GO TO 124
      J = I - 100
      GO TO 135
  124 IF(I.GT.200) GO TO 126
      L = I - 150
      M = 2*L - 1
      N = M + 100
      J = I - N
      GO TO 135
  126 IF(I.GT.250) GO TO 128
      J = I - 200
      GO TO 135
  128 L = I - 250
      M = 2*L - 1
      N = M + 200
      J = I - N
  135 KCON(I,1) = 10*J + 4
      KCON(I,2) = 10*K + 5
      COEF(I,1) = VALK3
  120 COEF(I,2) = VALK2
C
C   END OF DATA SETUP
C   NOW CREATE INPUT FOR ANALYZER
C
      WRITE(IOT,922)
  922 FORMAT(/,' ENTER NAME OF INPUT FILE, INCLUDING DRIVE',
     & ' DESIGNATION:')
      READ(IN,923) FNAME
  923 FORMAT(A12)
C
C     OPEN INPUT FILE
C
      OPEN(IWR,FILE=FNAME,STATUS='NEW',FORM='FORMATTED',IOSTAT=ICK)
      IF(ICK.GT.0)WRITE(IOT,924)
  924 FORMAT(/,' Trouble opening input file')
C
C     WRITE(IOT,925) FNAME
      WRITE(IOT,925)
C 925 FORMAT(/,' WRITING FILE :',A14)
  925 FORMAT(/,' WRITING FILE : TVSSI')
      WRITE(IWR,919) TITLE
  919 FORMAT(1X,A79)
      WRITE(IWR,908)(L1(I),I=1,8)
  908 FORMAT(9I4)
      WRITE(IWR,908)(L2(I),I=1,3)
      WRITE(IWR,908)(L3(I),I=1,6)
      WRITE(IWR,911) FL4(1),FL4(2),L4,FL4(3),FL4(4)
  911 FORMAT(F8.3,F8.5,I8,2F8.1)
      WRITE(IWR,912) SET2(1),SET2(2)
  912 FORMAT(2F8.0)
C
      DO 200  I = 1,50
      WRITE(IWR,913)(KCON(I,J),J=1,6)
  913 FORMAT(9I8)
      WRITE(IWR,914)(COEF(I,J),J=1,6)
  914 FORMAT(6F8.4)
  200 CONTINUE
C
      DO 250 I=51,300
      WRITE(IWR,913) KCON(I,1),KCON(I,2)
      WRITE(IWR,914) COEF(I,1),COEF(I,2)
  250 CONTINUE
C
      CLOSE(IWR,IOSTAT=ICK)
      IF(ICK.NE.0) WRITE(IOT,921)EEX
  921 FORMAT(' Trouble closing printer output file'  )
C
      STOP
      END
```

APPENDIX B

NTU53P COMPUTER GENERATED INPUT ANALYZER PROGRAM

```
C         THIS IS PROGRAM NTU53P
C
C
C         IT GENERATES AN INPUT FILE FOR THERMAL ANALYZER TO OBTAIN
C         1-5 EFFECTIVENESS-NTU RELATIONSHIP THAT IS NOT AVAILABLE IN
C         OPEN LITERATURE (3P MEANING THREE PARALLEL PASSES AND ONE
C         COUNTERFLOW PASS).
C
      DIMENSION COEF(300,6),KCON(300,6),L1(8),L2(3),L3(6),SET2(2),FL4(4)
C
      CHARACTER *79 TITLE
      CHARACTER *12 FNAME
C
      DATA IOT,IN,IPR,IWR/6,5,4,8/
C
C     OPEN PRINTER OUTPUT FILE
C
      OPEN(IPR,FILE='PRN',STATUS='NEW',FORM='FORMATTED',IOSTAT=ICK)
      IF(ICK.NE.0) WRITE(IOT,920)
  920 FORMAT(' Trouble opening printer output file' )
C
      WRITE(IOT,917)
  917 FORMAT(/' Input the title of this study - 79 columns only. ',
     & ' This title will appear',/,' at the top of every printed page',
     & ' of output:')
      READ(IN,918) TITLE
  918 FORMAT(A79)
C
      WRITE(IOT,901)
  901 FORMAT(/' INPUT HOT SIDE CAPACITY RATE:')
      READ(IN,902) CHOT
  902 FORMAT(BN,F10.0)
C
      WRITE(IOT,903)
  903 FORMAT(/' INPUT COLD SIDE CAPACITY RATE:')
      READ(IN,902) CCLD
C
      WRITE(IOT,904)
  904 FORMAT(/' INPUT OVERALL HEAT TRANSFER COEFFICIENT:')
      READ(IN,902) U
C
      WRITE(IOT,905)
  905 FORMAT(/' INPUT TOTAL HEAT TRANSFER SURFACE:')
      READ(IN,902) SURFTO
C
      WRITE(IOT,906)
  906 FORMAT(/' INPUT HOT SIDE INLET TEMPERATURE:')
      READ(IN,902) THOTIN
C
      WRITE(IOT,907)
  907 FORMAT(/' INPUT COLD SIDE INLET TEMPERATURE:')
      READ(IN,902) TCLDIN
C
      VALK1 = CHOT
      VALK2 = CCLD
      VALK3 = U*SURFTO/250.
      TINIT = 125.
C
C             FRONT END
C
      L1(1) = 300
      L1(2) = 2
      DO 10  I=3,8
   10 L1(I) = 0
C
      DO 20  I=1,3
   20 L2(I) = 0
C
      L3(1) = 300
      L3(2) = 50
      L3(3) = 6
      L3(4) = 2
      L3(5) = 4
      L3(6) = 6
C
      FL4(1) = .05
      FL4(2) = .66667
      FL4(3) = .8
      FL4(4) = TINIT
      L4 = 12
C
C   CONSTANT TEMPERATURES
C
      SET2(1) = THOTIN
      SET2(2) = TCLDIN
C
C   READY FOR INPUT SET 4
C
C   NODE 1
C
      KCON(1,1) = 514
      KCON(1,2) = 1504
      KCON(1,3) = 1514
      KCON(1,4) = 2504
      KCON(1,5) = 2514
      KCON(1,6) = 3015
      COEF(1,6) = VALK1
      DO 50 I = 1,5
   50 COEF(1,I) = VALK3
C
C     NODES 2 TO 50
C
      DO 75 I = 2,50
      J = I + 50
      K = 151 - I
      L = 150 + I
      N = I - 1
      M = 251 - I
      MM = 250 + I
      KCON(I,1) = 10*N + 5
      KCON(I,2) = 10*J + 4
      KCON(I,3) = 10*K + 4
      KCON(I,4) = 10*L + 4
      KCON(I,5) = 10 *M + 4
      KCON(I,6) = 10 *MM + 4
      COEF(I,1) = VALK1
      DO 80  II = 2,6
      COEF(I,II) = VALK3
   80 CONTINUE
   75 CONTINUE
C
C    NODE 51
C
      KCON(51,1) = 3025
      KCON(51,2) = 14
      COEF(51,1) = VALK2
      COEF(51,2) = VALK3
C
C     NODES 52 TO 300
C
      DO 120  I = 52,300
      K = I - 1
      IF(I.GT.100) GO TO 122
      J = I - 50
      GO TO 135
  122 IF(I.GT.150) GO TO 124
      L = I - 100
      M = 2*L - 1
      N = M + 50
      J = I - N
      GO TO 135
  124 IF(I.GT.200) GO TO 126
```

```
      J = I - 150
      GO TO 135
  126 IF(I.GT.250) GO TO 128
      L = I - 200
      M = 2*L - 1
      N=M+150
      J = I - N
      GO TO 135
  128 J = I - 250
  135 KCON(I,1) = 10*J + 4
      KCON(I,2) = 10*K + 5
      COEF(I,1) = VALK3
  120 COEF(I,2) = VALK2
C
C   END OF DATA SETUP
C   NOW CREATE INPUT FOR ANALYZER
C
      WRITE(IOT,922)
  922 FORMAT(/,' Enter name of input file, including drive',
     & ' DESIGNATION:')
      READ(IN,923) FNAME
  923 FORMAT(A12)
C
C     OPEN INPUT FILE
C
      OPEN(IWR,FILE=FNAME,STATUS='NEW',FORM='FORMATTED',IOSTAT=ICK)
      IF(ICK.GT.0)WRITE(IOT,924)
  924 FORMAT(/,' Trouble opening input file')
C
      WRITE(IOT,925) FNAME
  925 FORMAT(/,' Writing file :',A14)
      WRITE(IWR,919) TITLE
  919 FORMAT(1X,A79)
      WRITE(IWR,908)(L1(I),I=1,8)
  908 FORMAT(9I4)
      WRITE(IWR,908)(L2(I),I=1,3)
      WRITE(IWR,908)(L3(I),I=1,6)
      WRITE(IWR,911) FL4(1),FL4(2),L4,FL4(3),FL4(4)
  911 FORMAT(F8.3,F8.5,I8,F8.5,F8.2)
      WRITE(IWR,912) SET2(1),SET2(2)
  912 FORMAT(2F8.0)
C
      DO 200  I = 1,50
      WRITE(IWR,913)(KCON(I,J),J=1,6)
  913 FORMAT(9I8)
      WRITE(IWR,914)(COEF(I,J),J=1,6)
  914 FORMAT(6F8.4)
  200 CONTINUE
C
      DO 250 I=51,300
```

```
        WRITE(IWR,913) KCON(I,1),KCON(I,2)
        WRITE(IWR,914) COEF(I,1),COEF(I,2)
  250   CONTINUE
C
        CLOSE(IWR,IOSTAT=ICK)
        IF(ICK.NE.0) WRITE(IOT,921)EEX
  921   FORMAT(' Trouble closing printer output file'  )
C
        STOP
        END
```

APPENDIX C

SAMPLE OUTPUT FROM NTU53C COMPUTER INPUT ANALYZER PROGRAM

```
 R=0.5 NTU=1.0
 300   2   0   0   0   0   0   0
   0   0   0
 300  50   6   2   4   6
   0.050 0.66667      12     0.8   125.0
    200.    100.
    1004    1014    2004    2014    3004    3015
  0.5000  0.5000  0.5000  0.5000  0.5000250.0000
      15     994    1024    1994    2024    2994
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
      25     984    1034    1984    2034    2984
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
      35     974    1044    1974    2044    2974
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
      45     964    1054    1964    2054    2964
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
      55     954    1064    1954    2064    2954
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
      65     944    1074    1944    2074    2944
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
      75     934    1084    1934    2084    2934
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
      85     924    1094    1924    2094    2924
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
      95     914    1104    1914    2104    2914
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     105     904    1114    1904    2114    2904
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     115     894    1124    1894    2124    2894
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     125     884    1134    1884    2134    2884
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     135     874    1144    1874    2144    2874
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     145     864    1154    1864    2154    2864
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     155     854    1164    1854    2164    2854
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     165     844    1174    1844    2174    2844
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     175     834    1184    1834    2184    2834
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     185     824    1194    1824    2194    2824
```

```
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     195       814      1204      1814      2204      2814
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     205       804      1214      1804      2214      2804
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     215       794      1224      1794      2224      2794
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     225       784      1234      1784      2234      2784
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     235       774      1244      1774      2244      2774
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     245       764      1254      1764      2254      2764
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     255       754      1264      1754      2264      2754
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     265       744      1274      1744      2274      2744
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     275       734      1284      1734      2284      2734
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     285       724      1294      1724      2294      2724
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     295       714      1304      1714      2304      2714
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     305       704      1314      1704      2314      2704
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     315       694      1324      1694      2324      2694
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     325       684      1334      1684      2334      2684
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     335       674      1344      1674      2344      2674
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     345       664      1354      1664      2354      2664
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     355       654      1364      1654      2364      2654
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     365       644      1374      1644      2374      2644
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     375       634      1384      1634      2384      2634
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     385       624      1394      1624      2394      2624
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     395       614      1404      1614      2404      2614
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     405       604      1414      1604      2414      2604
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     415       594      1424      1594      2424      2594
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     425       584      1434      1584      2434      2584
250.0000    0.5000    0.5000    0.5000    0.0000    0.0000
     435       574      1444      1574      2444      2574
```

```
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     445     564    1454    1564    2454    2564
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     455     554    1464    1554    2464    2554
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     465     544    1474    1544    2474    2544
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     475     534    1484    1534    2484    2534
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     485     524    1494    1524    2494    2524
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
     495     514    1504    1514    2504    2514
250.0000  0.5000  0.5000  0.5000  0.0000  0.0000
    3025     504
125.0000  0.5000
     494     515
  0.5000125.0000
     484     525
  0.5000125.0000
     474     535
  0.5000125.0000
     464     545
  0.5000125.0000
     454     555
  0.5000125.0000
     444     565
  0.5000125.0000
     434     575
  0.5000125.0000
     424     585
  0.5000125.0000
     414     595
  0.5000125.0000
     404     605
  0.5000125.0000
     394     615
  0.5000125.0000
     384     625
  0.5000125.0000
     374     635
  0.5000125.0000
     364     645
  0.5000125.0000
     354     655
  0.5000125.0000
     344     665
  0.5000125.0000
     334     675
  0.5000125.0000
     324     685
```

```
0.5000125.0000
   314      695
0.5000125.0000
   304      705
0.5000125.0000
   294      715
0.5000125.0000
   284      725
0.5000125.0000
   274      735
0.5000125.0000
   264      745
0.5000125.0000
   254      755
0.5000125.0000
   244      765
0.5000125.0000
   234      775
0.5000125.0000
   224      785
0.5000125.0000
   214      795
0.5000125.0000
   204      805
0.5000125.0000
   194      815
0.5000125.0000
   184      825
0.5000125.0000
   174      835
0.5000125.0000
   164      845
0.5000125.0000
   154      855
0.5000125.0000
   144      865
0.5000125.0000
   134      875
0.5000125.0000
   124      885
0.5000125.0000
   114      895
0.5000125.0000
   104      905
0.5000125.0000
    94      915
0.5000125.0000
    84      925
0.5000125.0000
    74      935
```

```
0.5000125.0000
    64     945
0.5000125.0000
    54     955
0.5000125.0000
    44     965
0.5000125.0000
    34     975
0.5000125.0000
    24     985
0.5000125.0000
    14     995
0.5000125.0000
    14    1005
0.5000125.0000
    24    1015
0.5000125.0000
    34    1025
0.5000125.0000
    44    1035
0.5000125.0000
    54    1045
0.5000125.0000
    64    1055
0.5000125.0000
    74    1065
0.5000125.0000
    84    1075
0.5000125.0000
    94    1085
0.5000125.0000
   104    1095
0.5000125.0000
   114    1105
0.5000125.0000
   124    1115
0.5000125.0000
   134    1125
0.5000125.0000
   144    1135
0.5000125.0000
   154    1145
0.5000125.0000
   164    1155
0.5000125.0000
   174    1165
0.5000125.0000
   184    1175
0.5000125.0000
   194    1185
```

```
0.5000125.0000
   204    1195
0.5000125.0000
   214    1205
0.5000125.0000
   224    1215
0.5000125.0000
   234    1225
0.5000125.0000
   244    1235
0.5000125.0000
   254    1245
0.5000125.0000
   264    1255
0.5000125.0000
   274    1265
0.5000125.0000
   284    1275
0.5000125.0000
   294    1285
0.5000125.0000
   304    1295
0.5000125.0000
   314    1305
0.5000125.0000
   324    1315
0.5000125.0000
   334    1325
0.5000125.0000
   344    1335
0.5000125.0000
   354    1345
0.5000125.0000
   364    1355
0.5000125.0000
   374    1365
0.5000125.0000
   384    1375
0.5000125.0000
   394    1385
0.5000125.0000
   404    1395
0.5000125.0000
   414    1405
0.5000125.0000
   424    1415
0.5000125.0000
   434    1425
0.5000125.0000
   444    1435
```

```
0.5000125.0000
   454    1445
0.5000125.0000
   464    1455
0.5000125.0000
   474    1465
0.5000125.0000
   484    1475
0.5000125.0000
   494    1485
0.5000125.0000
   504    1495
0.5000125.0000
   504    1505
0.5000125.0000
   494    1515
0.5000125.0000
   484    1525
0.5000125.0000
   474    1535
0.5000125.0000
   464    1545
0.5000125.0000
   454    1555
0.5000125.0000
   444    1565
0.5000125.0000
   434    1575
0.5000125.0000
   424    1585
0.5000125.0000
   414    1595
0.5000125.0000
   404    1605
0.5000125.0000
   394    1615
0.5000125.0000
   384    1625
0.5000125.0000
   374    1635
0.5000125.0000
   364    1645
0.5000125.0000
   354    1655
0.5000125.0000
   344    1665
0.5000125.0000
   334    1675
0.5000125.0000
   324    1685
```

```
0.5000125.0000
   314    1695
0.5000125.0000
   304    1705
0.5000125.0000
   294    1715
0.5000125.0000
   284    1725
0.5000125.0000
   274    1735
0.5000125.0000
   264    1745
0.5000125.0000
   254    1755
0.5000125.0000
   244    1765
0.5000125.0000
   234    1775
0.5000125.0000
   224    1785
0.5000125.0000
   214    1795
0.5000125.0000
   204    1805
0.5000125.0000
   194    1815
0.5000125.0000
   184    1825
0.5000125.0000
   174    1835
0.5000125.0000
   164    1845
0.5000125.0000
   154    1855
0.5000125.0000
   144    1865
0.5000125.0000
   134    1875
0.5000125.0000
   124    1885
0.5000125.0000
   114    1895
0.5000125.0000
   104    1905
0.5000125.0000
    94    1915
0.5000125.0000
    84    1925
0.5000125.0000
    74    1935
```

```
0.5000125.0000
    64    1945
0.5000125.0000
    54    1955
0.5000125.0000
    44    1965
0.5000125.0000
    34    1975
0.5000125.0000
    24    1985
0.5000125.0000
    14    1995
0.5000125.0000
```

APPENDIX D

MODIFIED NTU53C PROGRAM USED TO RUN ON BATCH SYSTEM

```
//HESS  JOB (2054,0267),'NTU53C',CLASS=B
//*MAIN  ORG=NPGVM1.2054P,SYSTEM=SY2
//  EXEC FORTVCL,PARM.LKED='LIST,MAP'
//FORT.SYSIN DD *
C         THIS IS PROGRAM NTU53C
C
C
C         IT GENERATES AN INPUT FILE FOR THERMAL ANALYZER TO OBTAIN
C         1-5 EFFECTIVENESS-NTU RELATIONSHIP THAT IS NOT AVAILABLE IN
C         OPEN LITERATURE (3C MEANING TWO PARALLEL PASS AND THREE
C         COUNTERFLOW PASSES).
C
      INTEGER O,P
      DIMENSION COEF(300,6),KCON(300,6),L1(8),L2(3),L3(6),SET2(2),FL4(4)
     1,CHOT(22,22),CCLD(22,22),U(22,22),SURFTO(22,22),THOTIN(22,22),TCLD
     2IN(22,22),TITLE(22,22),FNAME(22,22)
C
      CHARACTER *25 TITLE
      CHARACTER *25 FNAME
C
      DO 7 O=1,21,2
      P=O+1
      READ(1,900) CHOT(O,1),CCLD(O,2),U(O,3),SURFTO(O,4),THOTIN(O,5),TCL
     1DIN(O,6)
900   FORMAT(2F10.0,F10.5,3F10.0)
      READ(1,918) TITLE(P,1),FNAME(P,2)
918   FORMAT(2A25)
C     OPEN OUTPUT FILE
C
      OPEN(2,FILE=FNAME(P,2),FORM='FORMATTED')
C
C
      VALK1 = CHOT(O,1)
      VALK2 = CCLD(O,2)
      VALK3 = U(O,3)*SURFTO(O,4)/250.
      TINIT = 125.
C
C             FRONT END
C
      L1(1) = 300
      L1(2) = 2
      DO 10  I=3,8
   10 L1(I) = 0
```

```
C
      DO 20  I=1,3
   20 L2(I) = 0
C
      L3(1) = 300
      L3(2) = 50
      L3(3) = 6
      L3(4) = 2
      L3(5) = 4
      L3(6) = 6
C
      FL4(1) = .05
      FL4(2) = .66667
      FL4(3) = .8
      FL4(4) = TINIT
      L4 = 12
C
C   CONSTANT TEMPERATURES
C
      SET2(1) = THOTIN(0,5)
      SET2(2) = TCLDIN(0,6)
C
C   READY FOR INPUT SET 4
C
C   NODE 1
C
      KCON(1,1) = 1004
      KCON(1,2) = 1014
      KCON(1,3) = 2004
      KCON(1,4) = 2014
      KCON(1,5) = 3004
      KCON(1,6) = 3015
      COEF(1,6) = VALK1
      DO 50 I = 1,5
   50 COEF(1,I) = VALK3
C
C   NODES 2 TO 50
C
      DO 75 I = 2,50
      J = 101 - I
      K = 100 + I
      L = 201 - I
      N = I - 1
      M=200+I
      MM=301-I
      KCON(I,1) = 10*N + 5
      KCON(I,2) = 10*J + 4
      KCON(I,3) = 10*K + 4
      KCON(I,4) = 10*L + 4
      KCON(I,5) = 10*M + 4
      KCON(I,6) = 10*MM + 4
      COEF(I,1) = VALK1
      DO 80  II = 2,6
      COEF(I,II) = VALK3
   80 CONTINUE
   75 CONTINUE
C
C  NODE 51
C
      KCON(51,1) = 3025
      KCON(51,2) = 504
      COEF(51,1) = VALK2
      COEF(51,2) = VALK3
C
C   NODES 52 TO 300
C
      DO 120  I = 52,300
      K = I - 1
      IF(I.GT.100) GO TO 122
      L = I - 50
      M = 2*L - 1
      J = I - M
      GO TO 135
  122 IF(I.GT.150) GO TO 124
      J = I - 100
      GO TO 135
  124 IF(I.GT.200) GO TO 126
      L = I - 150
      M = 2*L - 1
      N = M + 100
      J = I - N
      GO TO 135
  126 IF(I.GT.250) GO TO 128
      J = I - 200
      GO TO 135
  128 L = I - 250
      M = 2*L - 1
      N = M + 200
      J = I - N
  135 KCON(I,1) = 10*J + 4
      KCON(I,2) = 10*K + 5
      COEF(I,1) = VALK3
  120 COEF(I,2) = VALK2
C
C   END OF DATA SETUP
C   NOW CREATE INPUT FOR ANALYZER
C
      WRITE(2,919) TITLE(P,1)
  919 FORMAT(1X,A25)
      WRITE(2,908)(L1(I),I=1,8)
  908 FORMAT(9I4)
      WRITE(2,908)(L2(I),I=1,3)
      WRITE(2,908)(L3(I),I=1,6)
      WRITE(2,911) FL4(1),FL4(2),L4,FL4(3),FL4(4)
  911 FORMAT(F8.3,F8.5,I8,2F8.1)
      WRITE(2,912) SET2(1),SET2(2)
  912 FORMAT(2F8.0)
C
      DO 200  I = 1,50
      WRITE(2,913)(KCON(I,J),J=1,6)
  913 FORMAT(9I8)
      WRITE(2,914)(COEF(I,J),J=1,6)
  914 FORMAT(6F8.4)
  200 CONTINUE
C
      DO 250 I=51,300
      WRITE(2,913) KCON(I,1),KCON(I,2)
      WRITE(2,914) COEF(I,1),COEF(I,2)
  250 CONTINUE
C
    7 CONTINUE
      STOP
      END
/*
//LKED.SYSLMOD DD DISP=SHR,DSNAME=MSS.S2054.LOADLIB
//LKED.SYSIN   DD *
   NAME NTU53C(R)
/*
//
```

APPENDIX E

MASTER EXECUTION PROGRAM FOR NTU53C AND TVCOUNT

```
//HESS30C  JOB (2054,0267),'L E HESS',CLASS=P
//*MAIN  ORG=NPGVM1.2054P,LINES=(5000,W)
//*FORMAT PR,DDNAME=,DEST=LOCAL
//  EXEC FORTVG,PROG=NTU53C,LIB='MSS.S2054.LOADLIB'
//GO.TVSSIA    DD DISP=(NEW,PASS),DSN=&TVSSIA,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIB    DD DISP=(NEW,PASS),DSN=&TVSSIB,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIC    DD DISP=(NEW,PASS),DSN=&TVSSIC,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSID    DD DISP=(NEW,PASS),DSN=&TVSSID,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIE    DD DISP=(NEW,PASS),DSN=&TVSSIE,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIF    DD DISP=(NEW,PASS),DSN=&TVSSIF,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIG    DD DISP=(NEW,PASS),DSN=&TVSSIG,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIH    DD DISP=(NEW,PASS),DSN=&TVSSIH,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSII    DD DISP=(NEW,PASS),DSN=&TVSSII,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIJ    DD DISP=(NEW,PASS),DSN=&TVSSIJ,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIK    DD DISP=(NEW,PASS),DSN=&TVSSIK,
//             UNIT=SYSDA,SPACE=(CYL,(2,2)),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.FT01F001 DD *
      250.       75.0   0.25000       15.       200.       100.
NTU=0.05 R=0.30 COUNTER  TVSSIA
      250.       75.0   1.25000       15.       200.       100.
NTU=0.25 R=0.30 COUNTER  TVSSIB
      250.       75.0   2.50000       15.       200.       100.
```

```
NTU=0.50 R=0.30 COUNTER  TVSSIC
      250.      75.0   3.75000      15.      200.      100.
NTU=0.75 R=0.30 COUNTER  TVSSID
      250.      75.0   5.00000      15.      200.      100.
NTU=1.00 R=0.30 COUNTER  TVSSIE
      250.      75.0   6.25000      15.      200.      100.
NTU=1.25 R=0.30 COUNTER  TVSSIF
      250.      75.0   7.50000      15.      200.      100.
NTU=1.50 R=0.30 COUNTER  TVSSIG
      250.      75.0  10.00000      15.      200.      100.
NTU=2.00 R=0.30 COUNTER  TVSSIH
      250.      75.0  12.50000      15.      200.      100.
NTU=2.50 R=0.30 COUNTER  TVSSII
      250.      75.0  15.00000      15.      200.      100.
NTU=3.00 R=0.30 COUNTER  TVSSIJ
      250.      75.0  16.25000      15.      200.      100.
NTU=3.25 R=0.30 COUNTER  TVSSIK
/*
//STEPA  EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIA
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPB  EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIB
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPC  EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIC
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPD  EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSID
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
```

```
//STEPE   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIE
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPF   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIF
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPG   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIG
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPH   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIH
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPI   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSII
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPJ   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIJ
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPK   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

```
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIK
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//
```

APPENDIX F

MODIFIED NTU53P PROGRAM USED TO RUN ON BATCH SYSTEM

```
//HESS1 JOB (2054,0267),'NTU53P',CLASS=B
//*MAIN  ORG=NPGVM1.2054P,SYSTEM=SY2
//  EXEC FORTVCL,PARM.LKED='LIST,MAP'
//FORT.SYSIN DD *
C         THIS IS PROGRAM NTU53P
C
C
C         IT GENERATES AN INPUT FILE FOR THERMAL ANALYZER TO OBTAIN
C         1-5 EFFECTIVENESS-NTU RELATIONSHIP THAT IS NOT AVAILABLE IN
C         OPEN LITERATURE (3P MEANING THREE PARALLEL PASSES AND TWO
C         COUNTERFLOW PASS).
C
      INTEGER O,P
      DIMENSION COEF(300,6),KCON(300,6),L1(8),L2(3),L3(6),SET2(2),FL4(4)
     1,CHOT(22,22),CCLD(22,22),U(22,22),SURFTO(22,22),THOTIN(22,22),
     2TCLDIN(22,22),TITLE(22,22),FNAME(22,22)
C
      CHARACTER *25 TITLE
      CHARACTER *25 FNAME
C
      DO 7 O=1,21,2
      P=O+1
      READ(1,900) CHOT(O,1),CCLD(O,2),U(O,3),SURFTO(O,4),THOTIN(O,5),TCL
     1DIN(O,6)
  900 FORMAT(2F10.0,F10.5,3F10.0)
      READ(1,918) TITLE(P,1),FNAME(P,2)
918   FORMAT(2A25)
C     OPEN OUTPUT FILE
C
      OPEN(2,FILE=FNAME(P,2),FORM='FORMATTED')
C
      VALK1 = CHOT(O,1)
      VALK2 = CCLD(O,2)
      VALK3 = U(O,3)*SURFTO(O,4)/250.
      TINIT = 125.
C
C            FRONT END
C
      L1(1) = 300
      L1(2) = 2
      DO 10  I=3,8
   10 L1(I) = 0
C
```

```
      DO 20  I=1,3
   20 L2(I) = 0
C
      L3(1) = 300
      L3(2) = 50
      L3(3) = 6
      L3(4) = 2
      L3(5) = 4
      L3(6) = 6
C
      FL4(1) = .05
      FL4(2) = .66667
      FL4(3) = .8
      FL4(4) = TINIT
      L4 = 12
C
C    CONSTANT TEMPERATURES
C
      SET2(1) = THOTIN(0,5)
      SET2(2) = TCLDIN(0,6)
C
C    READY FOR INPUT SET 4
C
C    NODE 1
C
      KCON(1,1) = 514
      KCON(1,2) = 1504
      KCON(1,3) = 1514
      KCON(1,4) = 2004
      KCON(1,5) = 2014
      KCON(1,6) = 3015
      COEF(1,6) = VALK1
      DO 50 I = 1,5
   50 COEF(1,I) = VALK3
C
C    NODES 2 TO 50
C
      DO 75 I = 2,50
      J = I + 50
      K = 151 - I
      L = 150 + I
      N = I - 1
      M = 251 - I
      MM = 250 + I
      KCON(I,1) = 10*N + 5
      KCON(I,2) = 10*J + 4
      KCON(I,3) = 10*K + 4
      KCON(I,4) = 10*L + 4
      KCON(I,5) = 10 *M + 4
      KCON(I,6) = 10 *MM + 4
      COEF(I,1) = VALK1
      DO 80  II = 2,6
      COEF(I,II) = VALK3
   80 CONTINUE
   75 CONTINUE
C
C  NODE 51
C
      KCON(51,1) = 3025
      KCON(51,2) = 14
      COEF(51,1) = VALK2
      COEF(51,2) = VALK3
C
C   NODES 52 TO 300
C
      DO 120  I = 52,300
      K = I - 1
      IF(I.GT.100) GO TO 122
      J = I - 50
      GO TO 135
  122 IF(I.GT.150) GO TO 124
      L = I - 100
      M = 2*L - 1
      N = M + 50
      J = I - N
      GO TO 135
  124 IF(I.GT.200) GO TO 126
      J = I - 150
      GO TO 135
  126 IF(I.GT.250) GO TO 128
      L = I - 200
      M = 2*L - 1
      N = M + 150
      J = I - N
      GO TO 135
  128 J = I - 250
  135 KCON(I,1) = 10*J + 4
      KCON(I,2) = 10*K + 5
      COEF(I,1) = VALK3
  120 COEF(I,2) = VALK2
C
C   END OF DATA SETUP
C   NOW CREATE INPUT FOR ANALYZER
C
C
      WRITE(2,919) TITLE(P,1)
  919 FORMAT(1X,A25)
      WRITE(2,908)(L1(I),I=1,8)
  908 FORMAT(9I4)
      WRITE(2,908)(L2(I),I=1,3)
      WRITE(2,908)(L3(I),I=1,6)
      WRITE(2,911) FL4(1),FL4(2),L4,FL4(3),FL4(4)
  911 FORMAT(F8.3,F8.5,I8,F8.5,F8.2)
      WRITE(2,912) SET2(1),SET2(2)
  912 FORMAT(2F8.0)
C
      DO 200  I = 1,50
      WRITE(2,913)(KCON(I,J),J=1,6)
  913 FORMAT(9I8)
      WRITE(2,914)(COEF(I,J),J=1,6)
  914 FORMAT(6F8.4)
  200 CONTINUE
C
      DO 250 I=51,300
      WRITE(2,913) KCON(I,1),KCON(I,2)
      WRITE(2,914) COEF(I,1),COEF(I,2)
  250 CONTINUE
C
    7 CONTINUE
      STOP
      END
/*
//LKED.SYSLMOD DD DISP=SHR,DSNAME=MSS.S2054.LOADLIB
//LKED.SYSIN   DD *
   NAME NTU53P(R)
/*
//
```

APPENDIX G

MASTER EXECUTION PROGRAM FOR NTU53C AND TVCOUNT

```
//HESS30P  JOB (2054,0267),'L E HESS',CLASS=P
//*MAIN  ORG=NPGVM1.2054P,LINES=(5000,W)
//*FORMAT PR,DDNAME=,DEST=LOCAL
//  EXEC FORTVG,PROG=NTU53P,LIB='MSS.S2054.LOADLIB'
//GO.TVSSIL   DD DISP=(NEW,PASS),DSN=&TVSSIL,
//            UNIT=SYSDA,SPACE=(CYL,(2,2)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIM   DD DISP=(NEW,PASS),DSN=&TVSSIM,
//            UNIT=SYSDA,SPACE=(CYL,(2,2)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIN   DD DISP=(NEW,PASS),DSN=&TVSSIN,
//            UNIT=SYSDA,SPACE=(CYL,(2,2)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIO   DD DISP=(NEW,PASS),DSN=&TVSSIO,
//            UNIT=SYSDA,SPACE=(CYL,(2,2)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIP   DD DISP=(NEW,PASS),DSN=&TVSSIP,
//            UNIT=SYSDA,SPACE=(CYL,(2,2)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIQ   DD DISP=(NEW,PASS),DSN=&TVSSIQ,
//            UNIT=SYSDA,SPACE=(CYL,(2,2)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIR   DD DISP=(NEW,PASS),DSN=&TVSSIR,
//            UNIT=SYSDA,SPACE=(CYL,(2,2)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIS   DD DISP=(NEW,PASS),DSN=&TVSSIS,
//            UNIT=SYSDA,SPACE=(CYL,(2,2)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIT   DD DISP=(NEW,PASS),DSN=&TVSSIT,
//            UNIT=SYSDA,SPACE=(CYL,(2,2)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIU   DD DISP=(NEW,PASS),DSN=&TVSSIU,
//            UNIT=SYSDA,SPACE=(CYL,(2,2)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.TVSSIV   DD DISP=(NEW,PASS),DSN=&TVSSIV,
//            UNIT=SYSDA,SPACE=(CYL,(2,2)),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=6160)
//GO.FT01F001 DD *
      250.      75.0   0.25000       15.      200.       100.
NTU=0.05 R=0.30 PARALLEL TVSSIL
      250.      75.0   1.25000       15.      200.       100.
NTU=0.25 R=0.30 PARALLEL TVSSIM
      250.      75.0   2.50000       15.      200.       100.
```

```
NTU=0.50 R=0.30 PARALLEL TVSSIN
      250.      75.0   3.75000       15.      200.      100.
NTU=0.75 R=0.30 PARALLEL TVSSIO
      250.      75.0   5.00000       15.      200.      100.
NTU=1.00 R=0.30 PARALLEL TVSSIP
      250.      75.0   6.25000       15.      200.      100.
NTU=1.25 R=0.30 PARALLEL TVSSIQ
      250.      75.0   7.50000       15.      200.      100.
NTU=1.50 R=0.30 PARALLEL TVSSIR
      250.      75.0  10.00000       15.      200.      100.
NTU=2.00 R=0.30 PARALLEL TVSSIS
      250.      75.0  12.50000       15.      200.      100.
NTU=2.50 R=0.30 PARALLEL TVSSIT
      250.      75.0  15.00000       15.      200.      100.
NTU=3.00 R=0.30 PARALLEL TVSSIU
      250.      75.0  16.25000       15.      200.      100.
NTU=3.25 R=0.30 PARALLEL TVSSIV
/*
//STEPL  EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIL
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPM  EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIM
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPN  EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIN
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPO  EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIO
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
```

```
//STEPP   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIP
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPQ   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIQ
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPR   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIR
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPS   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIS
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPT   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIT
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPU   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIU
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//STEPV   EXEC FORTVG,PROG=TVSSIAC,LIB='MSS.S2054.LOADLIB'
//GO.FT01F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
```

```
//GO.FT02F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT09F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT10F001 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//GO.FT03F001 DD DUMMY
//GO.FT04F001 DD DISP=(OLD,DELETE),UNIT=SYSDA,DSN=&TVSSIV
//GO.FT08F001 DD SYSOUT=A,DCB=RECFM=FBA
//
```

APPENDIX H

1-5:3P EFFECTIVENESS VS. N_{tu} GRAPHS AT VARIOUS R VALUES

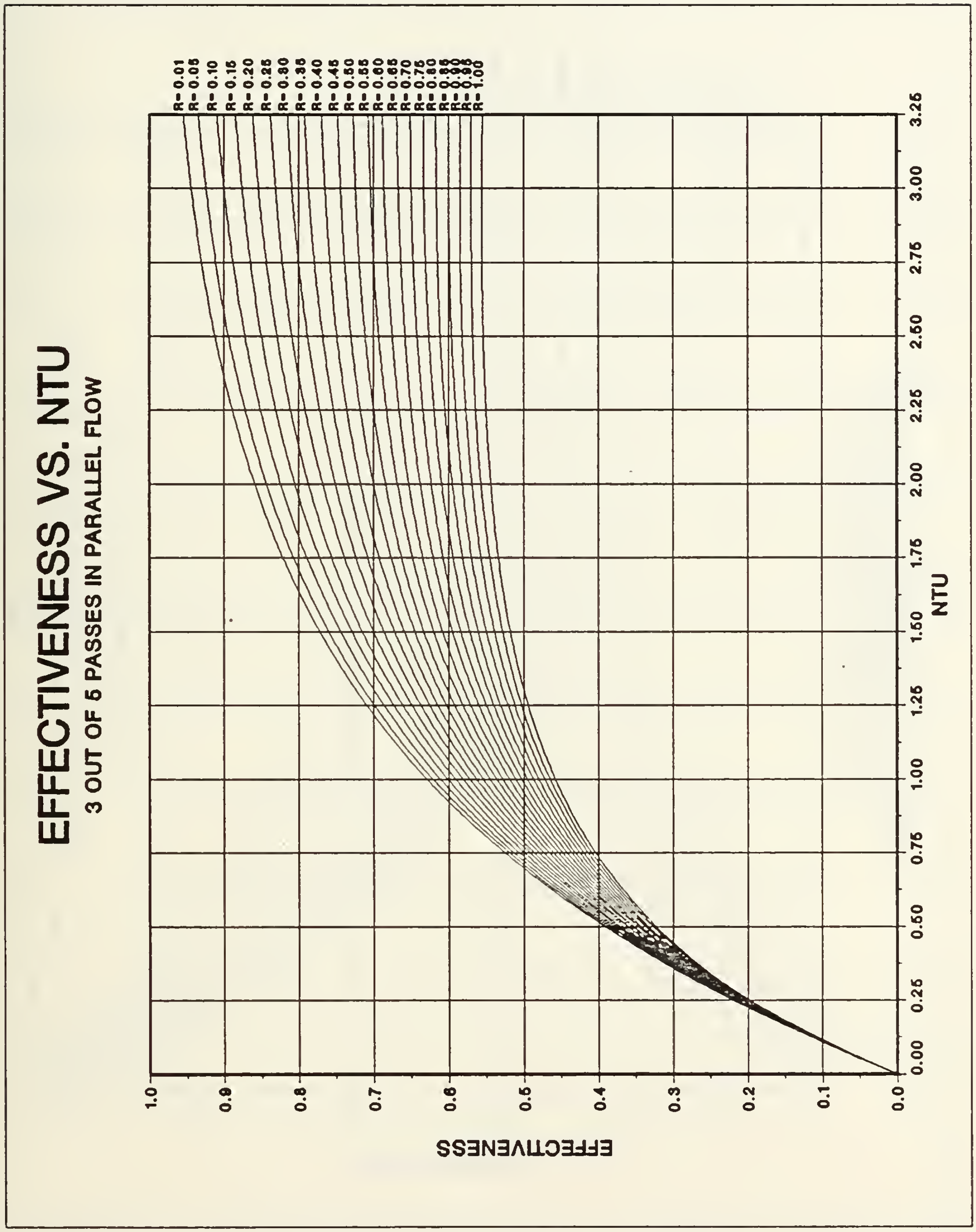

Figure H.1 1-5:3P Effectiveness vs. N_{tu} over Range of R from 0.01 to 1.0

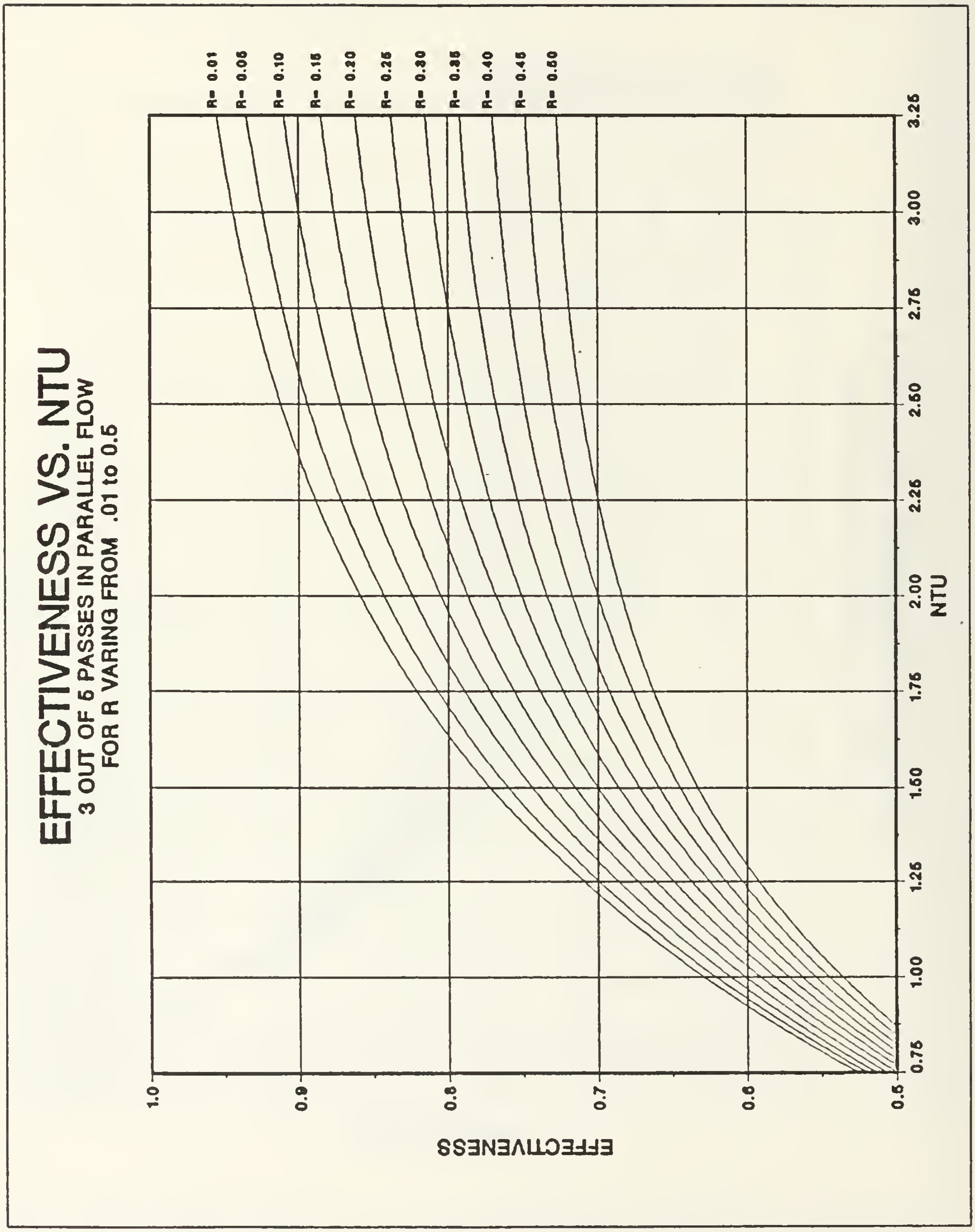

Figure H.2 1-5:3P Effectiveness vs. N_{tu} over Range of R from 0.01 to 0.5

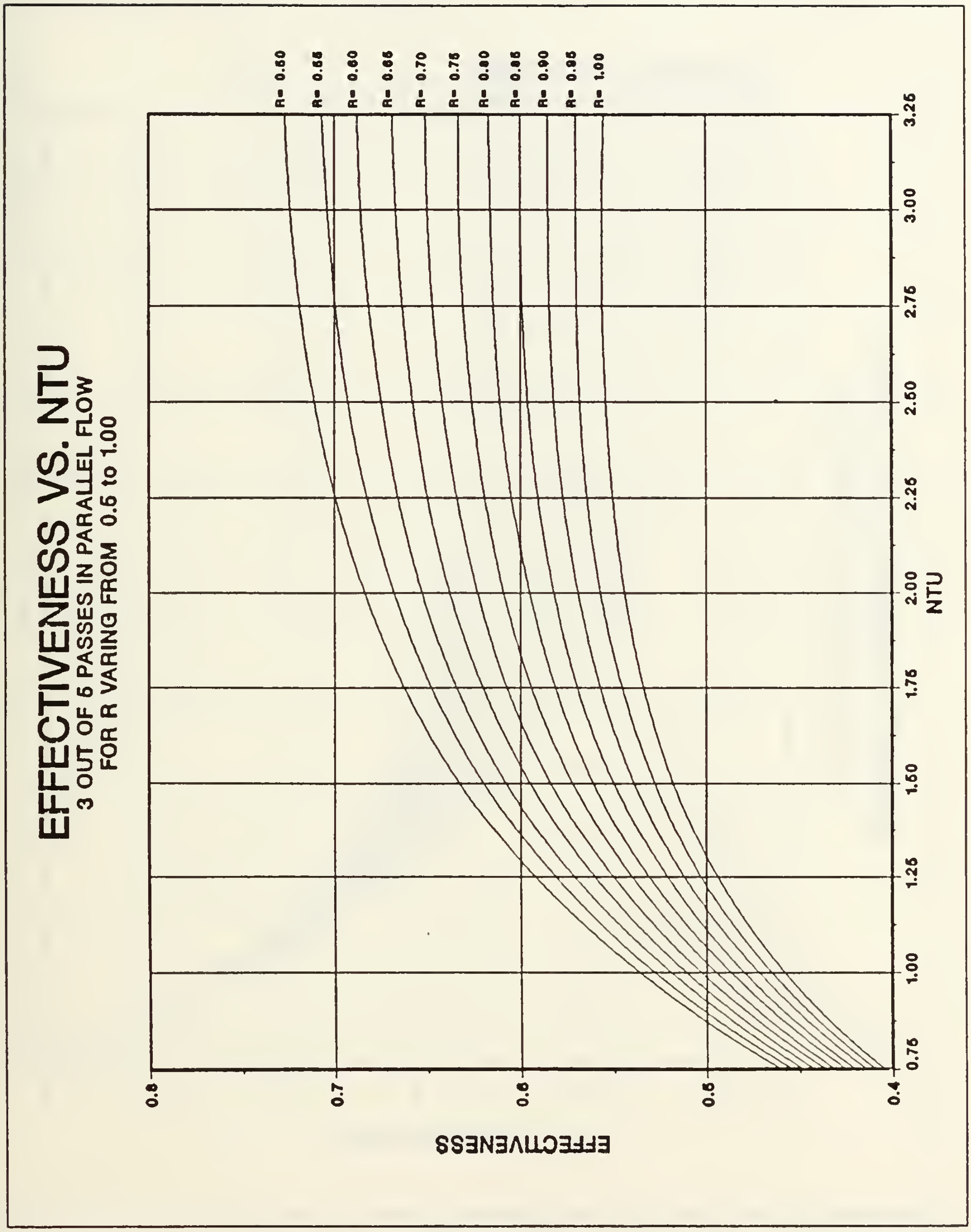

Figure H.3 1-5:3P Effectiveness vs. N_{tu} over Range of R from 0.5 to 1.00

APPENDIX I

1-5:3C EFFECTIVENESS VS. N_{tu} GRAPHS AT VARIOUS R VALUES

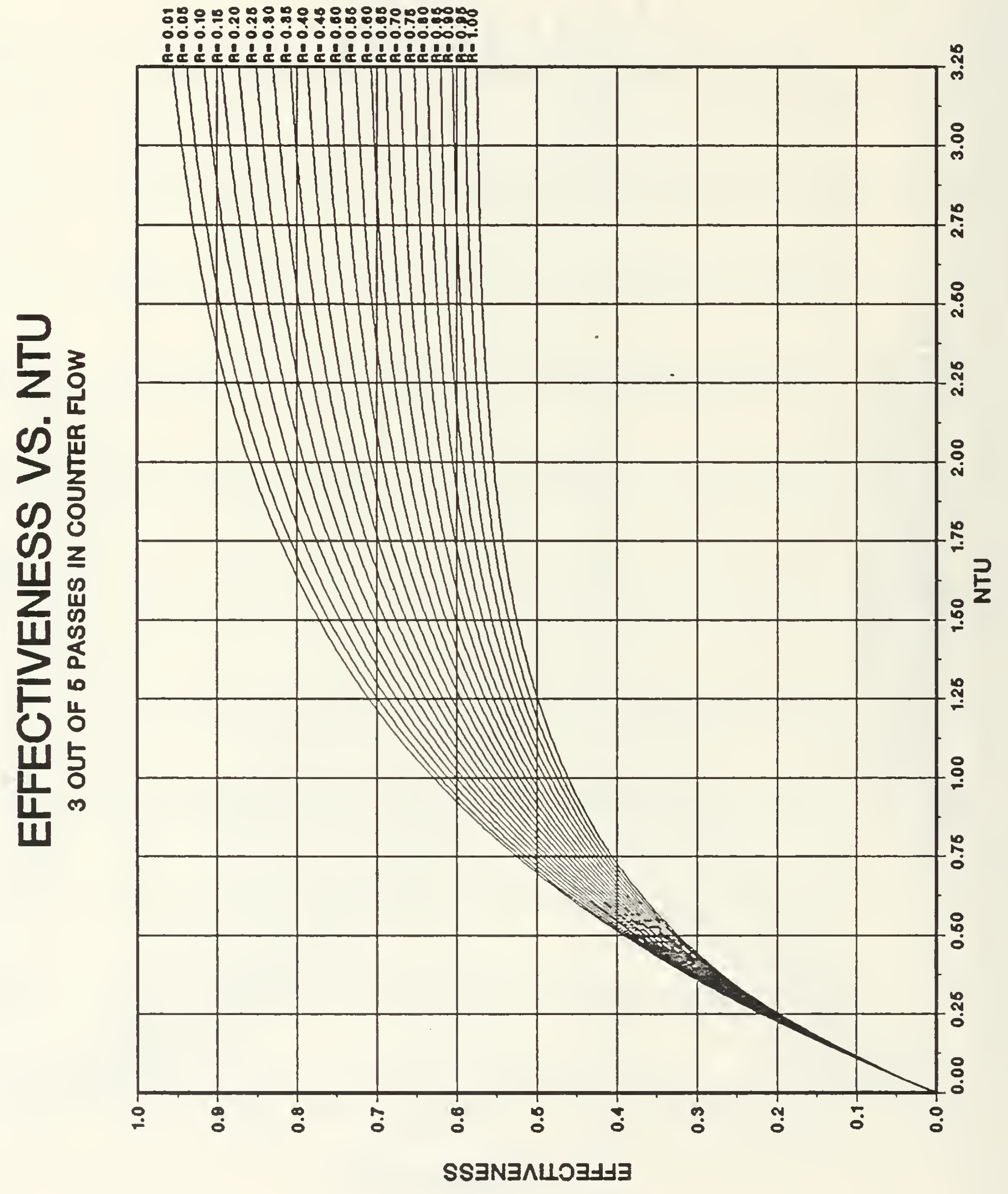

Figure I.1 1-5:3C Effectiveness vs. N_{tu} over Range of R from 0.01 to 1.0

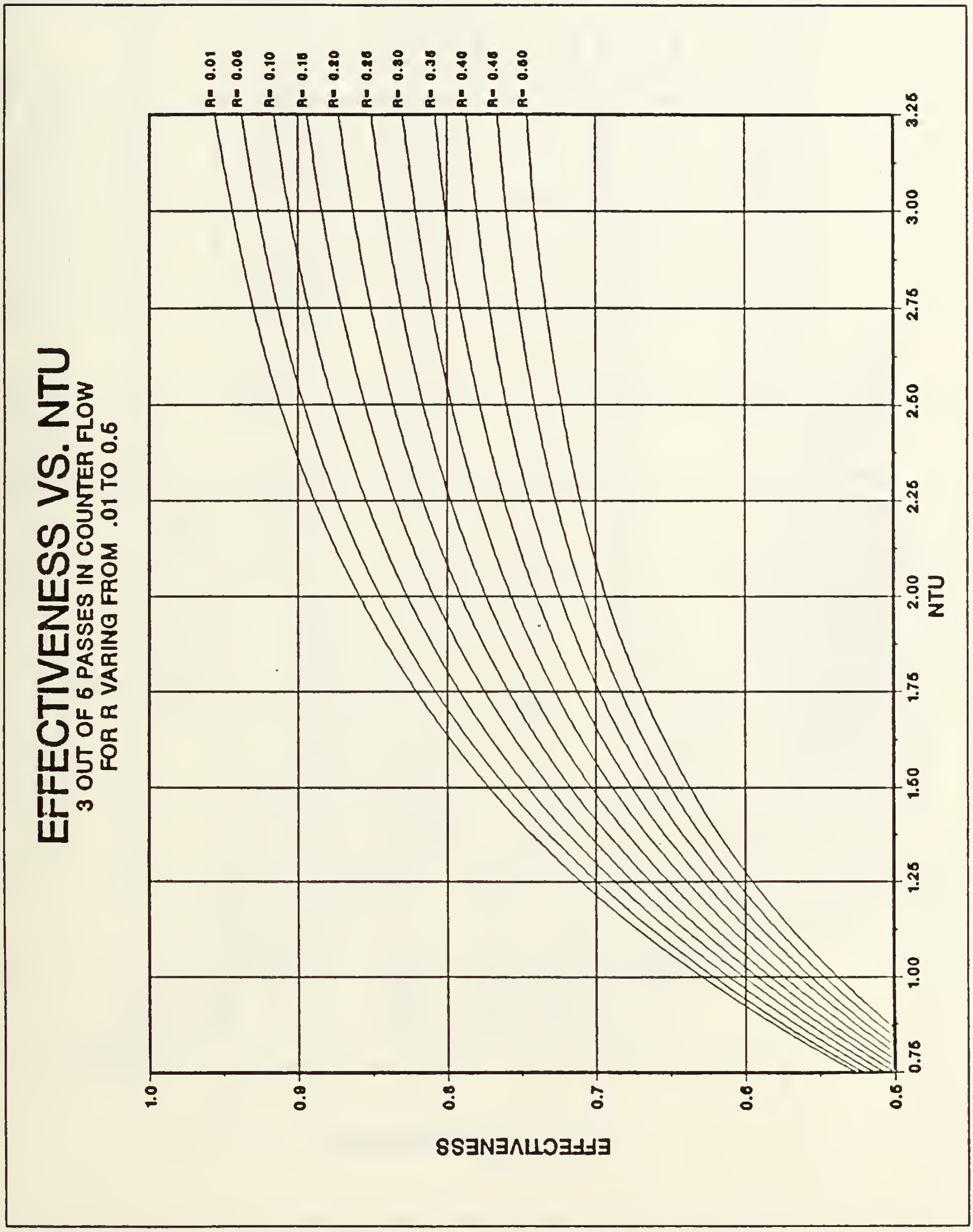

Figure I.2 1-5:3C Effectiveness vs. N_{tu} over Range of R from 0.01 to 0.5

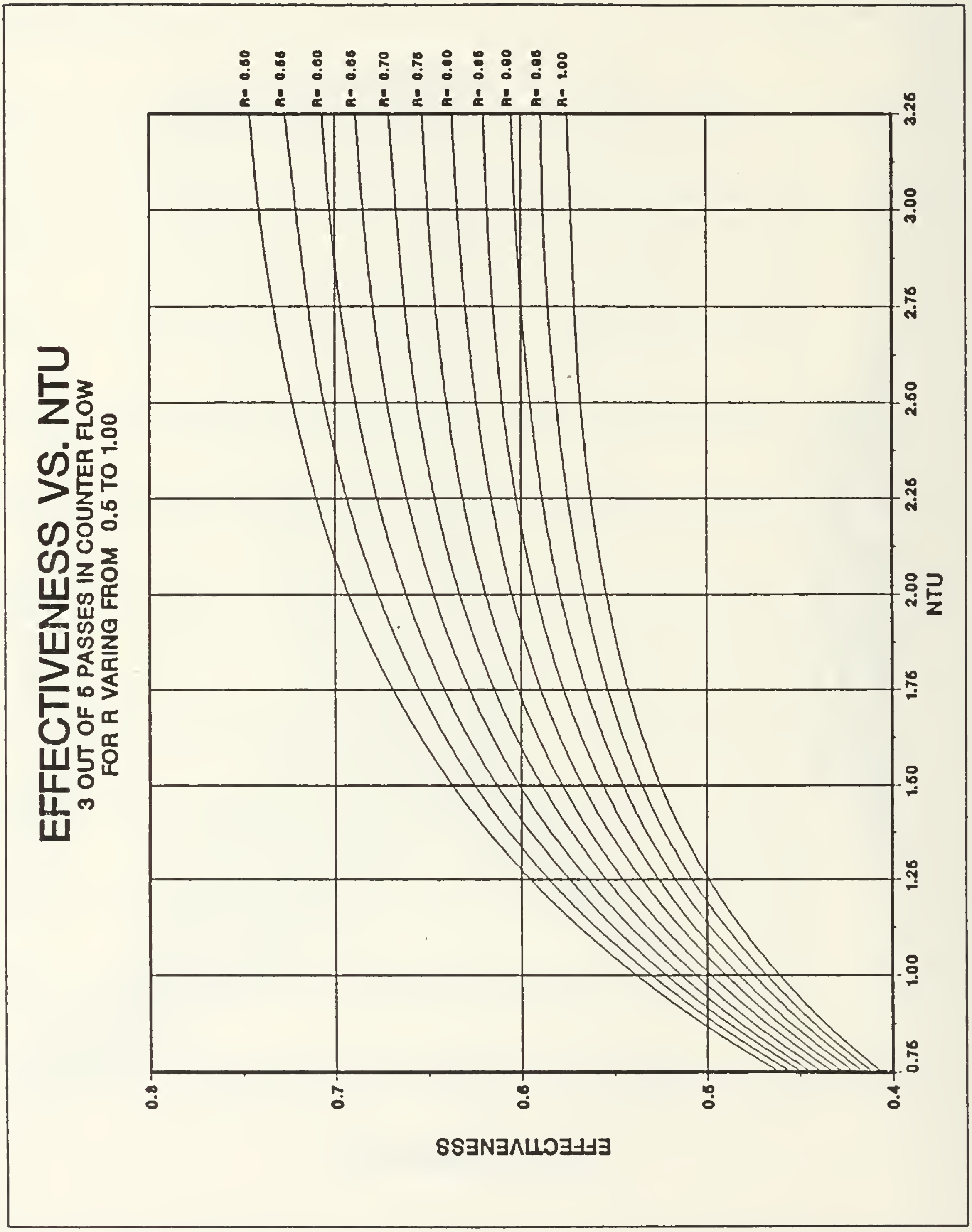

Figure I.3 1-5:3C Effectiveness vs. N_{tu} over Range of R from 0.5 to 1.00

APPENDIX J

EFFECTIVENESS COMPUTER GENERATED ANALYZER PROGRAM

```
C***********************************************************************
C                        EFFECTIVENESS PROGRAM
C                          LCDR L. E. HESS
C                           12 AUGUST 1985
C***********************************************************************
C
C  THIS PROGRAM CONTAINS THE 1-5 POLYNOMIAL APPROXIMATION COEFFICIENTS.
C  BY ENTERING THE VALUES FOR NTU, R AND THE TYPE OF HEAT EXCHANGER, THE
C  EFFECTIVENESS VALUES ARE READILY OBTAINED.  BY EITHER USING THE 1-5
C  COUNTER OR PARALLEL COEFFICIENTS DIRECTLY OR BY LINEAR INTERPOLATION
C
      REAL NTU,E,E1,E2,R,R1,R2
      CHARACTER*10 EXCH
      WRITE(6,10)
10    FORMAT (/' INPUT THE TYPE OF HEAT EXCHANGER (PARALLEL OR COUNTER)'
     *)
      READ(5,20) EXCH
20    FORMAT(A10)
      WRITE(6,30)
30    FORMAT (/' INPUT NUMBER OF TRANSFER UNITS (NTU)')
      READ(5,40) NTU
40    FORMAT (BN,F10.0)
      WRITE(6,50)
50    FORMAT (/' INPUT CAPACITY RATE RATIO (R). R MUST BE GREATER THAN O
     *R EQUAL TO 0.01',/,' AND LESS THAN OR EQUAL TO 1.00')
      READ(5,40) R
      IF (EXCH.EQ.'COUNTER') THEN
         IF (R.GE.0.01) THEN
         IF (R.LE.0.05) THEN
            R1=0.01
            R2=0.05
            E1=.00057463+0.99093*NTU-0.47938*NTU**2+0.13879*NTU**3-0.023
     *      075*NTU**4+0.0016779*NTU**5
            E2=.00079100+0.98831*NTU-0.49081*NTU**2+0.14654*NTU**3-0.025
     *      068*NTU**4+0.0018647*NTU**5
            IF(R.EQ.0.01) THEN
              WRITE (6,60) E1
60            FORMAT (/' EFFECTIVENESS =',F7.4)
              ELSE IF (R.EQ.0.05) THEN
                      WRITE (6,60) E2
              ELSE
                   CALL INTERP (R,R1,R2,E1,E2,E)
                   WRITE (6,60) E
            ENDIF
         ENDIF
         ENDIF
         IF (R.GT.0.05) THEN
         IF (R.LE.0.10) THEN
            R1=0.05
            R2=0.10
            E1=.00079100+0.98831*NTU-0.49081*NTU**2+0.14654*NTU**3-0.025
     *      068*NTU**4+0.0018647*NTU**5
            E2=.00108210+0.98622*NTU-0.50716*NTU**2+0.15773*NTU**3-0.028
     *      045*NTU**4+0.0021547*NTU**5
            IF(R.EQ.0.10) THEN
              WRITE (6,60) E2
              ELSE
                   CALL INTERP (R,R1,R2,E1,E2,E)
                   WRITE (6,60) E
            ENDIF
         ENDIF
         ENDIF
         IF (R.GT.0.10) THEN
         IF (R.LE.0.15) THEN
            R1=0.10
            R2=0.15
            E1=.00108210+0.98622*NTU-0.50716*NTU**2+0.15773*NTU**3-0.028
     *      045*NTU**4+0.0021547*NTU**5
            E2=.00125311+0.98238*NTU-0.52072*NTU**2+0.16779*NTU**3-0.030
     *      947*NTU**4+0.0024635*NTU**5
            IF(R.EQ.0.15) THEN
              WRITE (6,60) E2
              ELSE
                   CALL INTERP (R,R1,R2,E1,E2,E)
                   WRITE (6,60) E
            ENDIF
         ENDIF
         ENDIF
         IF (R.GT.0.15) THEN
         IF (R.LE.0.20) THEN
            R1=0.15
            R2=0.20
            E1=.00125311+0.98238*NTU-0.52072*NTU**2+0.16779*NTU**3-0.030
     *      947*NTU**4+0.0024635*NTU**5
            E2=.00142506+0.97866*NTU-0.53259*NTU**2+0.17545*NTU**3-0.032
     *      772*NTU**4+0.0026203*NTU**5
            IF(R.EQ.0.20) THEN
              WRITE (6,60) E2
              ELSE
                   CALL INTERP (R,R1,R2,E1,E2,E)
                   WRITE (6,60) E
            ENDIF
         ENDIF
```

```
      ENDIF
      IF (R.GT.0.20) THEN
      IF (R.LE.0.25) THEN
         R1=0.20
         R2=0.25
         E1=.00142506+0.97866*NTU-0.53259*NTU**2+0.17545*NTU**3-0.032
     *   772*NTU**4+0.0026203*NTU**5
         E2=.00133145+0.97670*NTU-0.54709*NTU**2+0.18482*NTU**3-0.035
     *   072*NTU**4+0.0028213*NTU**5
         IF(R.EQ.0.25) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.25) THEN
      IF (R.LE.0.30) THEN
         R1=0.25
         R2=0.30
         E1=.00133145+0.97670*NTU-0.54709*NTU**2+0.18482*NTU**3-0.035
     *   072*NTU**4+0.0028213*NTU**5
         E2=.00151081+0.97405*NTU-0.56166*NTU**2+0.19541*NTU**3-0.038
     *   055*NTU**4+0.0031252*NTU**5
         IF(R.EQ.0.30) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.30) THEN
      IF (R.LE.0.35) THEN
         R1=0.30
         R2=0.35
         E1=.00151081+0.97405*NTU-0.56166*NTU**2+0.19541*NTU**3-0.038
     *   055*NTU**4+0.0031252*NTU**5
         E2=.00174198+0.97091*NTU-0.57527*NTU**2+0.20542*NTU**3-0.040
     *   903*NTU**4+0.0034186*NTU**5
         IF(R.EQ.0.35) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.35) THEN
```

```
    IF (R.LE.0.40) THEN
       R1=0.35
       R2=0.40
       E1=.00174198+0.97091*NTU-0.57527*NTU**2+0.20542*NTU**3-0.040
*      903*NTU**4+0.0034186*NTU**5
       E2=.00184400+0.96879*NTU-0.59067*NTU**2+0.21718*NTU**3-0.044
*      469*NTU**4+0.0038122*NTU**5
       IF(R.EQ.0.40) THEN
         WRITE (6,60) E2
         ELSE
              CALL INTERP (R,R1,R2,E1,E2,E)
              WRITE (6,60) E
       ENDIF
    ENDIF
    ENDIF
    IF (R.GT.0.40) THEN
    IF (R.LE.0.45) THEN
       R1=0.40
       R2=0.45
       E1=.00184400+0.96879*NTU-0.59067*NTU**2+0.21718*NTU**3-0.044
*      469*NTU**4+0.0038122*NTU**5
       E2=.00213420+0.96459*NTU-0.60073*NTU**2+0.22416*NTU**3-0.046
*      295*NTU**4+0.0039861*NTU**5
       IF(R.EQ.0.45) THEN
         WRITE (6,60) E2
         ELSE
              CALL INTERP (R,R1,R2,E1,E2,E)
              WRITE (6,60) E
       ENDIF
    ENDIF
    ENDIF
    IF (R.GT.0.45) THEN
    IF (R.LE.0.50) THEN
       R1=0.45
       R2=0.50
       E1=.00213420+0.96459*NTU-0.60073*NTU**2+0.22416*NTU**3-0.046
*      295*NTU**4+0.0039861*NTU**5
       E2=.00230808+0.96140*NTU-0.61293*NTU**2+0.23322*NTU**3-0.048
*      933*NTU**4+0.0042669*NTU**5
       IF(R.EQ.0.50) THEN
         WRITE (6,60) E2
         ELSE
              CALL INTERP (R,R1,R2,E1,E2,E)
              WRITE (6,60) E
       ENDIF
    ENDIF
    ENDIF
    IF (R.GT.0.50) THEN
    IF (R.LE.0.55) THEN
       R1=0.50
```

```
         R2=0.55
         E1=.00230808+0.96140*NTU-0.61293*NTU**2+0.23322*NTU**3-0.048
  *      933*NTU**4+0.0042669*NTU**5
         E2=.00257953+0.95713*NTU-0.62234*NTU**2+0.23996*NTU**3-0.050
  *      762*NTU**4+0.0044499*NTU**5
         IF(R.EQ.0.55) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.55) THEN
      IF (R.LE.0.60) THEN
         R1=0.55
         R2=0.60
         E1=.00257953+0.95713*NTU-0.62234*NTU**2+0.23996*NTU**3-0.050
  *      762*NTU**4+0.0044499*NTU**5
         E2=.00274755+0.95336*NTU-0.63230*NTU**2+0.24706*NTU**3-0.052
  *      681*NTU**4+0.0046391*NTU**5
         IF(R.EQ.0.60) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.60) THEN
      IF (R.LE.0.65) THEN
         R1=0.60
         R2=0.65
         E1=.00274755+0.95336*NTU-0.63230*NTU**2+0.24706*NTU**3-0.052
  *      681*NTU**4+0.0046391*NTU**5
         E2=.00305249+0.94903*NTU-0.64122*NTU**2+0.25366*NTU**3-0.054
  *      530*NTU**4+0.0048293*NTU**5
         IF(R.EQ.0.65) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.65) THEN
      IF (R.LE.0.70) THEN
         R1=0.65
         R2=0.70
         E1=.00305249+0.94903*NTU-0.64122*NTU**2+0.25366*NTU**3-0.054
```

```
     *     530*NTU**4+0.0048293*NTU**5
           E2=.00319573+0.94580*NTU-0.65184*NTU**2+0.26137*NTU**3-0.056
     *     656*NTU**4+0.0050422*NTU**5
           IF(R.EQ.0.70) THEN
             WRITE (6,60) E2
             ELSE
                  CALL INTERP (R,R1,R2,E1,E2,E)
                  WRITE (6,60) E
           ENDIF
        ENDIF
        ENDIF
        IF (R.GT.0.70) THEN
        IF (R.LE.0.75) THEN
           R1=0.70
           R2=0.75
           E1=.00319573+0.94580*NTU-0.65184*NTU**2+0.26137*NTU**3-0.056
     *     656*NTU**4+0.0050422*NTU**5
           E2=.00339253+0.94193*NTU-0.66116*NTU**2+0.26833*NTU**3-0.058
     *     615*NTU**4+0.0052431*NTU**5
           IF(R.EQ.0.75) THEN
             WRITE (6,60) E2
             ELSE
                  CALL INTERP (R,R1,R2,E1,E2,E)
                  WRITE (6,60) E
           ENDIF
        ENDIF
        ENDIF
        IF (R.GT.0.75) THEN
        IF (R.LE.0.80) THEN
           R1=0.75
           R2=0.80
           E1=.00339253+0.94193*NTU-0.66116*NTU**2+0.26833*NTU**3-0.058
     *     615*NTU**4+0.0052431*NTU**5
           E2=.00363419+0.93822*NTU-0.67053*NTU**2+0.27531*NTU**3-0.060
     *     558*NTU**4+0.0054393*NTU**5
           IF(R.EQ.0.80) THEN
             WRITE (6,60) E2
             ELSE
                  CALL INTERP (R,R1,R2,E1,E2,E)
                  WRITE (6,60) E
           ENDIF
        ENDIF
        ENDIF
        IF (R.GT.0.80) THEN
        IF (R.LE.0.85) THEN
           R1=0.80
           R2=0.85
           E1=.00363419+0.93822*NTU-0.67053*NTU**2+0.27531*NTU**3-0.060
     *     558*NTU**4+0.0054393*NTU**5
           E2=.00382659+0.93407*NTU-0.67841*NTU**2+0.28109*NTU**3-0.062
```

```
  *      088*NTU**4+0.0055850*NTU**5
         IF(R.EQ.0.85) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.85) THEN
      IF (R.LE.0.90) THEN
         R1=0.85
         R2=0.90
         E1=.00382659+0.93407*NTU-0.67841*NTU**2+0.28109*NTU**3-0.062
  *      088*NTU**4+0.0055850*NTU**5
         E2=.00398221+0.93052*NTU-0.68748*NTU**2+0.28799*NTU**3-0.064
  *      051*NTU**4+0.0057874*NTU**5
         IF(R.EQ.0.90) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.90) THEN
      IF (R.LE.0.95) THEN
         R1=0.90
         R2=0.95
         E1=.00398221+0.93052*NTU-0.68748*NTU**2+0.28799*NTU**3-0.064
  *      051*NTU**4+0.0057874*NTU**5
         E2=.00426945+0.92609*NTU-0.69496*NTU**2+0.29381*NTU**3-0.065
  *      653*NTU**4+0.0059442*NTU**5
         IF(R.EQ.0.95) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.95) THEN
      IF (R.LE.1.00) THEN
         R1=0.95
         R2=1.00
         E1=.00426945+0.92609*NTU-0.69496*NTU**2+0.29381*NTU**3-0.065
  *      653*NTU**4+0.0059442*NTU**5
         E2=.00440670+0.92259*NTU-0.70365*NTU**2+0.30046*NTU**3-0.067
  *      514*NTU**4+0.0061309*NTU**5
         IF(R.EQ.1.00) THEN
         WRITE (6,60) E2
         ELSE
              CALL INTERP (R,R1,R2,E1,E2,E)
              WRITE (6,60) E
       ENDIF
    ENDIF
    ENDIF
 ELSE IF (EXCH.EQ.'PARALLEL') THEN
    IF (R.GE.0.01) THEN
    IF (R.LE.0.05) THEN
       R1=0.01
       R2=0.05
       E1=.00052233+0.99152*NTU-0.48055*NTU**2+0.13953*NTU**3-0.023
*      283*NTU**4+0.0016989*NTU**5
       E2=.00081101+0.98817*NTU-0.49062*NTU**2+0.14604*NTU**3-0.024
*      883*NTU**4+0.0018421*NTU**5
       IF(R.EQ.0.01) THEN
         WRITE (6,60) E1
         ELSE IF (R.EQ.0.05) THEN
                 WRITE (6,60) E2
         ELSE
              CALL INTERP (R,R1,R2,E1,E2,E)
              WRITE (6,60) E
       ENDIF
    ENDIF
    ENDIF
    IF (R.GT.0.05) THEN
    IF (R.LE.0.10) THEN
       R1=0.05
       R2=0.10
       E1=.00081101+0.98817*NTU-0.49062*NTU**2+0.14604*NTU**3-0.024
*      883*NTU**4+0.0018421*NTU**5
       E2=.00110555+0.98599*NTU-0.50718*NTU**2+0.15721*NTU**3-0.027
*      865*NTU**4+0.0021337*NTU**5
       IF(R.EQ.0.10) THEN
         WRITE (6,60) E2
         ELSE
              CALL INTERP (R,R1,R2,E1,E2,E)
              WRITE (6,60) E
       ENDIF
    ENDIF
    ENDIF
    IF (R.GT.0.10) THEN
    IF (R.LE.0.15) THEN
       R1=0.10
       R2=0.15
       E1=.00110555+0.98599*NTU-0.50718*NTU**2+0.15721*NTU**3-0.027
*      865*NTU**4+0.0021337*NTU**5
       E2=.00114518+0.98309*NTU-0.52246*NTU**2+0.16811*NTU**3-0.030
*      968*NTU**4+0.0024604*NTU**5
```

```
            IF(R.EQ.0.15) THEN
              WRITE (6,60) E2
              ELSE
                   CALL INTERP (R,R1,R2,E1,E2,E)
                   WRITE (6,60) E
            ENDIF
         ENDIF
         ENDIF
         IF (R.GT.0.15) THEN
         IF (R.LE.0.20) THEN
            R1=0.15
            R2=0.20
            E1=.00114518+0.98309*NTU-0.52246*NTU**2+0.16811*NTU**3-0.030
     *      968*NTU**4+0.0024604*NTU**5
            E2=.00125862+0.97914*NTU-0.53367*NTU**2+0.17485*NTU**3-0.032
     *      443*NTU**4+0.0025741*NTU**5
            IF(R.EQ.0.20) THEN
              WRITE (6,60) E2
              ELSE
                   CALL INTERP (R,R1,R2,E1,E2,E)
                   WRITE (6,60) E
            ENDIF
         ENDIF
         ENDIF
         IF (R.GT.0.20) THEN
         IF (R.LE.0.25) THEN
            R1=0.20
            R2=0.25
            E1=.00125862+0.97914*NTU-0.53367*NTU**2+0.17485*NTU**3-0.032
     *      443*NTU**4+0.0025741*NTU**5
            E2=.00129233+0.97701*NTU-0.54864*NTU**2+0.18464*NTU**3-0.034
     *      902*NTU**4+0.0027929*NTU**5
            IF(R.EQ.0.25) THEN
              WRITE (6,60) E2
              ELSE
                   CALL INTERP (R,R1,R2,E1,E2,E)
                   WRITE (6,60) E
            ENDIF
         ENDIF
         ENDIF
         IF (R.GT.0.25) THEN
         IF (R.LE.0.30) THEN
            R1=0.25
            R2=0.30
            E1=.00129233+0.97701*NTU-0.54864*NTU**2+0.18464*NTU**3-0.034
     *      902*NTU**4+0.0027929*NTU**5
            E2=.00143113+0.97558*NTU-0.56664*NTU**2+0.19807*NTU**3-0.038
     *      885*NTU**4+0.0032195*NTU**5
            IF(R.EQ.0.30) THEN
              WRITE (6,60) E2
```

```
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.30) THEN
      IF (R.LE.0.35) THEN
         R1=0.30
         R2=0.35
         E1=.00143113+0.97558*NTU-0.56664*NTU**2+0.19807*NTU**3-0.038
  *      885*NTU**4+0.0032195*NTU**5
         E2=.00169053+0.97206*NTU-0.57958*NTU**2+0.20740*NTU**3-0.041
  *      500*NTU**4+0.0034854*NTU**5
         IF(R.EQ.0.35) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.35) THEN
      IF (R.LE.0.40) THEN
         R1=0.35
         R2=0.40
         E1=.00169053+0.97206*NTU-0.57958*NTU**2+0.20740*NTU**3-0.041
  *      500*NTU**4+0.0034854*NTU**5
         E2=.00176316+0.96998*NTU-0.59515*NTU**2+0.21912*NTU**3-0.045
  *      054*NTU**4+0.0038805*NTU**5
         IF(R.EQ.0.40) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.40) THEN
      IF (R.LE.0.45) THEN
         R1=0.40
         R2=0.45
         E1=.00176316+0.96998*NTU-0.59515*NTU**2+0.21912*NTU**3-0.045
  *      054*NTU**4+0.0038805*NTU**5
         E2=.00210897+0.96559*NTU-0.60498*NTU**2+0.22583*NTU**3-0.046
  *      805*NTU**4+0.0040487*NTU**5
         IF(R.EQ.0.45) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.45) THEN
      IF (R.LE.0.50) THEN
         R1=0.45
         R2=0.50
         E1=.00210897+0.96559*NTU-0.60498*NTU**2+0.22583*NTU**3-0.046
     *   805*NTU**4+0.0040487*NTU**5
         E2=.00230583+0.96163*NTU-0.61546*NTU**2+0.23327*NTU**3-0.048
     *   857*NTU**4+0.0042594*NTU**5
         IF(R.EQ.0.50) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.50) THEN
      IF (R.LE.0.55) THEN
         R1=0.50
         R2=0.55
         E1=.00230583+0.96163*NTU-0.61546*NTU**2+0.23327*NTU**3-0.048
     *   857*NTU**4+0.0042594*NTU**5
         E2=.00259035+0.95726*NTU-0.62480*NTU**2+0.23985*NTU**3-0.050
     *   625*NTU**4+0.0044354*NTU**5
         IF(R.EQ.0.55) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
      ENDIF
      ENDIF
      IF (R.GT.0.55) THEN
      IF (R.LE.0.60) THEN
         R1=0.55
         R2=0.60
         E1=.00259035+0.95726*NTU-0.62480*NTU**2+0.23985*NTU**3-0.050
     *   625*NTU**4+0.0044354*NTU**5
         E2=.00277374+0.95331*NTU-0.63434*NTU**2+0.24638*NTU**3-0.052
     *   303*NTU**4+0.0045925*NTU**5
         IF(R.EQ.0.60) THEN
           WRITE (6,60) E2
           ELSE
                CALL INTERP (R,R1,R2,E1,E2,E)
                WRITE (6,60) E
         ENDIF
```

```
    ENDIF
    ENDIF
    IF (R.GT.0.60) THEN
    IF (R.LE.0.65) THEN
       R1=0.60
       R2=0.65
       E1=.00277374+0.95331*NTU-0.63434*NTU**2+0.24638*NTU**3-0.052
*      303*NTU**4+0.0045925*NTU**5
       E2=.00291148+0.95025*NTU-0.64574*NTU**2+0.25460*NTU**3-0.054
*      593*NTU**4+0.0048260*NTU**5
       IF(R.EQ.0.65) THEN
         WRITE (6,60) E2
         ELSE
              CALL INTERP (R,R1,R2,E1,E2,E)
              WRITE (6,60) E
       ENDIF
    ENDIF
    ENDIF
    IF (R.GT.0.65) THEN
    IF (R.LE.0.70) THEN
       R1=0.65
       R2=0.70
       E1=.00291148+0.95025*NTU-0.64574*NTU**2+0.25460*NTU**3-0.054
*      593*NTU**4+0.0048260*NTU**5
       E2=.00322294+0.94575*NTU-0.65400*NTU**2+0.26039*NTU**3-0.056
*      053*NTU**4+0.0049575*NTU**5
       IF(R.EQ.0.70) THEN
         WRITE (6,60) E2
         ELSE
              CALL INTERP (R,R1,R2,E1,E2,E)
              WRITE (6,60) E
       ENDIF
    ENDIF
    ENDIF
    IF (R.GT.0.70) THEN
    IF (R.LE.0.75) THEN
       R1=0.70
       R2=0.75
       E1=.00322294+0.94575*NTU-0.65400*NTU**2+0.26039*NTU**3-0.056
*      053*NTU**4+0.0049575*NTU**5
       E2=.00338351+0.94231*NTU-0.66425*NTU**2+0.26788*NTU**3-0.058
*      110*NTU**4+0.0051617*NTU**5
       IF(R.EQ.0.75) THEN
         WRITE (6,60) E2
         ELSE
              CALL INTERP (R,R1,R2,E1,E2,E)
              WRITE (6,60) E
       ENDIF
    ENDIF
    ENDIF
```

```
         IF (R.GT.0.75) THEN
         IF (R.LE.0.80) THEN
            R1=0.75
            R2=0.80
            E1=.00338351+0.94231*NTU-0.66425*NTU**2+0.26788*NTU**3-0.058
     *      110*NTU**4+0.0051617*NTU**5
            E2=.00360234+0.93852*NTU-0.67369*NTU**2+0.27495*NTU**3-0.060
     *      079*NTU**4+0.0053606*NTU**5
            IF(R.EQ.0.80) THEN
              WRITE (6,60) E2
              ELSE
                   CALL INTERP (R,R1,R2,E1,E2,E)
                   WRITE (6,60) E
            ENDIF
         ENDIF
         ENDIF
         IF (R.GT.0.80) THEN
         IF (R.LE.0.85) THEN
            R1=0.80
            R2=0.85
            E1=.00360234+0.93852*NTU-0.67369*NTU**2+0.27495*NTU**3-0.060
     *      079*NTU**4+0.0053606*NTU**5
            E2=.00382374+0.93414*NTU-0.68130*NTU**2+0.28045*NTU**3-0.061
     *      487*NTU**4+0.0054884*NTU**5
            IF(R.EQ.0.85) THEN
              WRITE (6,60) E2
              ELSE
                   CALL INTERP (R,R1,R2,E1,E2,E)
                   WRITE (6,60) E
            ENDIF
         ENDIF
         ENDIF
         IF (R.GT.0.85) THEN
         IF (R.LE.0.90) THEN
            R1=0.85
            R2=0.90
            E1=.00382374+0.93414*NTU-0.68130*NTU**2+0.28045*NTU**3-0.061
     *      487*NTU**4+0.0054884*NTU**5
            E2=.00391805+0.93133*NTU-0.69231*NTU**2+0.28893*NTU**3-0.063
     *      953*NTU**4+0.0057472*NTU**5
            IF(R.EQ.0.90) THEN
              WRITE (6,60) E2
              ELSE
                   CALL INTERP (R,R1,R2,E1,E2,E)
                   WRITE (6,60) E
            ENDIF
         ENDIF
         ENDIF
         IF (R.GT.0.90) THEN
         IF (R.LE.0.95) THEN
```

```
       R1=0.90
       R2=0.95
       E1=.00391805+0.93133*NTU-0.69231*NTU**2+0.28893*NTU**3-0.063
*      953*NTU**4+0.0057472*NTU**5
       E2=.00418844+0.92723*NTU-0.70050*NTU**2+0.29533*NTU**3-0.065
*      754*NTU**4+0.0059298*NTU**5
       IF(R.EQ.0.95) THEN
         WRITE (6,60) E2
         ELSE
              CALL INTERP (R,R1,R2,E1,E2,E)
              WRITE (6,60) E
       ENDIF
    ENDIF
    ENDIF
    IF (R.GT.0.95) THEN
    IF (R.LE.1.00) THEN
       R1=0.95
       R2=1.00
       E1=.00418844+0.92723*NTU-0.70050*NTU**2+0.29533*NTU**3-0.065
*      754*NTU**4+0.0059298*NTU**5
       E2=.00435186+0.92385*NTU-0.71022*NTU**2+0.30310*NTU**3-0.068
*      037*NTU**4+0.0061690*NTU**5
       IF(R.EQ.1.00) THEN
         WRITE (6,60) E2
         ELSE
              CALL INTERP (R,R1,R2,E1,E2,E)
              WRITE (6,60) E
       ENDIF
    ENDIF
    ENDIF
 ENDIF
 STOP
 END
 SUBROUTINE INTERP (R,R1,R2,E1,E2,E)
 E=0.0
 E=(E1-E2)*(R-R2)/(R1-R2)+E2
 RETURN
 END
```

LIST OF REFERENCES

1. O'Hare, M. S., The Effectiveness of Heat Exchangers with One Shell Pass and Three Tube Passes, M. S. Thesis, Naval Postgraduate School, Monterey, CA, June 1985.

2. Nagle, W. M., "Mean Temperature Differences in Multipass Heat Exchangers," Industrial and Engineering Chemistry, Vol. 25, 1933.

3. Davis, F. K., Ross Heater and Manufacturing Company Bulletin, 350, 72, 1931.

4. Kern, D. Q., Process Heat Transfer, McGraw-Hill Book Company, New York, NY, 1950.

5. Standards of the Tubular Heat Exchanger Manufacturers Association (TEMA), Sixth Edition, New York, NY, 1978.

6. Underwood, A. J. V., "The Calculation of the Mean Temperature Difference in Multipass Heat Exchangers," Journal of Institute of Petroleum Technologists, Vol. 20, 1934.

7. Bowman, R. A., "Mean Temperature Difference Correction in Multipass Exchangers," Industrial and Engineering Chemistry, Vol. 28, 1936, pp. 541-544.

8. Kraus, A. D. and Kern, D. Q., ASME Paper 65-HT-18, presented at the ASME-AIChE Heat Transfer Conference, Los Angeles, CA, 1965.

9. Bowman, R. A., Mueller, A. C. and Nagle, W. M., "Mean Temperature Difference in Design," Trans. ASME, Vol. 62, 1940.

10. Ten Broeck, H. A., "Multipass Exchanger Calculations," Industrial and Engineering Chemistry, Vol. 30, 1938, pp. 1041-1042.

11. Kays, W. M. and London, A. L., Compact Heat Exchangers, Third Edition, McGraw-Hill Book Company, New York, NY, 1985.

12. Fischer, F. K., "Mean Temperature Difference Correction in Multipass Exchangers," Industrial and Engineering Chemistry, Vol. 30, 1938, pp. 377-383.

13. Stevens, R. A., and others, "Mean-Temperature Difference in One, Two and Three-Pass Crossflow Heat Exchangers," Trans. ASME, Vol. 79, 1957, pp. 287.

14. Kern, D. Q. and Kraus, A. D., Extended Surface Heat Transfer, McGraw-Hill Book Company, New York, NY, 1972.

15. Gerald, C. F., Applied Numerical Analysis, 2nd edition, Addison-Wesley Publishing Company, Reading, Massachusetts, 1980.

16. Non-International Mathematical and Statistical Library (IMSL), Curvefit Program, Computer Program on Library IBM 3033 AP System, Naval Postgraduate School, Monterey, CA, 1983.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5100	2
3.	Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5100	1
4.	Mr. Lawrence E. Hess, Jr. 515 Dreshertown Rd. Fort Washington, PA 19034	1
5.	Daniel Library The Citadel Charleston, SC 29409 Attn: Mr. Herb Nath	1
6.	LCDR Lawrence E. Hess, USN 10 Gillespie Lane Monterey, CA 93940	2
7.	Professor A. D. Kraus, Code 69Ks Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93943-5100	2
8.	LCDR Mark S. O'Hare, USN 107 Peyton Drive Chicago Hgts, IL 60411	1

1994 /.14

214331

Thesis
H5232 Hess
c.1 The effectiveness of
heat exchangers with
one shell pass and five
tube passes.

214331

Thesis
H5232 Hess
c.1 The effectiveness of
heat exchangers with
one shell pass and five
tube passes.

AMERICAN BINDERIES
2

thesH5232
The effectiveness of heat exchangers wit
3 2768 000 64934 7
DUDLEY KNOX LIBRARY